

Cómo cocinar tu contrato ágil

Xavier Albaladejo

xavi.albaladejo@gmail.com

Resumen. Este artículo muestra una visión estructurada de diferentes modalidades de contratos ágiles (en función de qué si se fijan o no las variables alcance, coste o plazos, desde contratos cerrados hasta *Time & Materials* o servicios, pasando por diferentes posibilidades de pago), asimilándolos a diferentes maneras de cocinar y sus posibles guarniciones (cláusulas adicionales que puede ser interesante incluir en el contrato en función del contexto). Se indica cuándo puede ser más conveniente utilizar cada tipo de contrato y qué se puede hacer si el cliente ya ha fijado algunas de las variables. Asimismo, se resalta la importancia de explicitar en el contrato las reglas que facilitarán la colaboración entre las partes. El artículo parte del Agile Contracts Primer, de Tom Arbogast, Craig Larman y Bas Vodde.

Palabras clave: contratos ágiles, contratos cerrados, contratos de coste objetivo, contratos progresivos, finalización anticipada, *money for nothing*, *change for free*, pago por iteración, límites de pago.

1 Fundamentos

1.1 Acuerdos y Reglas

Los principales objetivos de un contrato son:

- Establecer un acuerdo entre cliente y proveedor sobre el servicio a proporcionar. En este acuerdo las dos partes “ganan”, hay un beneficio mutuo en la realización del servicio: tanto el cliente como el proveedor deben poder satisfacer sus expectativas de producto, de alcance, de plazos, económicas, etc.
- Definir las reglas que regirán el servicio y que permitirán mantener este acuerdo durante el proyecto.

Contrato = acuerdo = win-win ⇔ Reglas



Fig. 1. Triángulo de hierro.

En línea con que todas las partes involucradas tienen que ganar, los riesgos del proyecto (respecto a expectativas, retrasos, costes superiores a lo estimado, etc.) deberían ser compartidos, de manera que no se entre en un juego de ocultación de información o sobreprotección y, por el contrario, se facilite la colaboración y creación de sinergias para obtener mejores soluciones.

1.2 Equipos de Alto Rendimiento que Incluyen al Cliente

Uno de los principales objetivos de Agile es crear equipos de alto rendimiento, bien engranados y con fuertes sinergias. Esto no se consigue de un día para otro, necesita de trabajo conjunto durante suficiente tiempo, colaboración entre los miembros del equipo y soporte mutuo para conseguir resultados, fomentando así la confianza y transparencia. En Agile una parte fundamental del equipo es el cliente, por lo que el alto rendimiento se consigue creando relaciones a medio-largo plazo entre cliente y proveedor que lleven a resultados mejores proyecto a proyecto. Por el contrario, no se busca hacer un proyecto de cualquier manera, en la que alguna de las partes pierda para acabar rompiendo un primer vínculo sin haber conseguido ese equipo engranado.

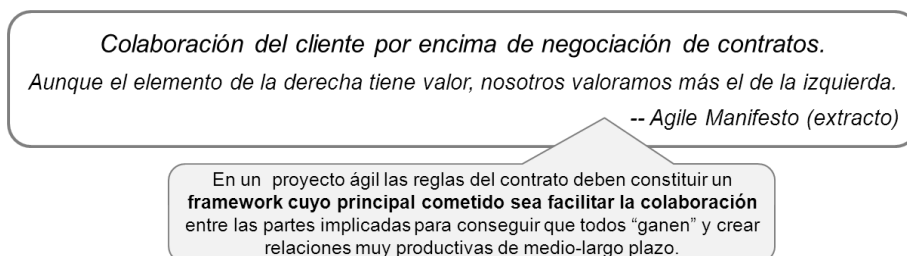


Fig. 2. El contrato como *framework* de colaboración.

Si en algún momento durante el proyecto hay que recurrir al contrato que se firmó al inicio, posiblemente esta sea una mala señal de que las cosas están empezando a ir mal y que la confianza se está deteriorando. Curiosamente, en este punto las partes suelen intentar colaborar más unas con otras para que la relación no se rompa y el proyecto finalice exitosamente, creando planes de mejora (como en una retrospectiva).

1.3 Complejidad, División, Feedback Rápido, Reflexión y Cambios

La complejidad en un proyecto o servicio se puede analizar considerando los siguientes factores:

- Las personas que participan, el conocimiento que tengan del producto a desarrollar, de cómo desarrollarlo, su capacidad para tomar decisiones, su estilo e idiosincrasia, etc.
- El entorno organizativo y el proceso de trabajo, especialización por funciones con grandes trasposos de información, burocracia, responsabilidad dispersa, exceso de jerarquía, etc.
- La tecnología y herramientas utilizadas, el conocimiento que exista de ellas, su madurez y estabilidad, etc.
- Los requisitos del proyecto, si están suficientemente maduros como para ser desarrollados inmediatamente, si son complejos por sí mismos, etc.

La complejidad global de un proyecto puede ser muy grande si se trabaja con todos estos factores de simultáneamente, dado que su combinación hace mucho más difícil su gestión. El desarrollo *waterfall* (cascada), sigue un planteamiento similar: en inicio del proyecto se intenta disponer de todos los requisitos detallados (abordar toda la complejidad funcional de una vez) y posteriormente se realiza el desarrollo considerando todas las necesidades tecnológicas de manera simultánea.

Una estrategia para gestionar los riesgos derivados de trabajar con mucha complejidad es dividir el producto en partes pequeñas, para así a sólo afrontar, paso a paso, una porción más reducida de esta complejidad, escogiendo y planificando qué riesgos se quiere mitigar en cada momento, por ejemplo:

- Riesgos tecnológicos: anticipar las partes en que se van a abordar y hacer pruebas de concepto lo más espartanas posibles (*spikes*) antes de su desarrollo.
- Riesgos de requisitos inmaduros: no ejecutar al inicio dado que habrá hipótesis que fallarán, dar tiempo a que el cliente para que vaya entendiendo cuál es el producto que necesita, mediante la inspección de las partes ya desarrolladas (normalmente las que le aportan más valor, que son las que deberían estar más claras).

Este planteamiento se alinea con otro de los principales objetivos de Agile, tener feedback rápido de si lo que se está haciendo es lo que se espera, a nivel de expectativas, velocidad de desarrollo, etc., De este modo también se facilita al cliente y al proveedor el ir aprendiendo en ciclos cortos sobre el producto que se está desarrollando, reflexionar sobre los procesos de trabajo, relaciones entre personas, impedimentos técnicos, etc., y actuar en consecuencia (control empírico). Es decir, la complejidad inherente a un proyecto lleva a aceptar que, de manera natural, habrá cambios. Los cambios son necesarios.

1.4 Los Proyectos son Infinitos

Los proyectos (y, en especial, los productos) pueden crecer y alargarse tanto como como poderosa sea la imaginación de sus *sponsors*, su capacidad de innovación o las necesidades de sus usuarios o consumidores finales. Es por ello que es conveniente conocer cuáles son los objetivos del proyecto que más ayudarán al usuario final y, en el inicio de un proyecto, saber dónde tendrá sentido cortar y fasear para poder disponer de soluciones suficientemente buenas a tiempo.

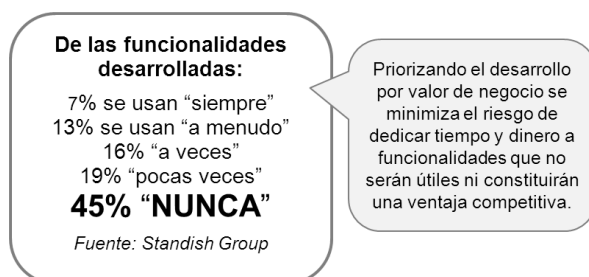


Fig. 3. La mitad de lo que se desarrolla no sirve para nada. Estudio de Standish Group [7].

1.5. Waterfall es Arriesgado

La elaboración de un contrato para un proyecto *waterfall* (el método de desarrollo tradicional, una secuencia de actividades de requisitos, análisis, diseño, construcción y pruebas) suele ser una optimización local que disminuye el resultado global, lo realmente importante, el producto que espera el cliente. Por ejemplo, suelen establecer:

- Una firma inicial de requisitos o del análisis funcional (con el objetivo de asegurar que se desarrollará todo lo esperado, de la manera especificada en estas primeras etapas del proyecto).
- Una revisión intermedia de sub-productos (esencialmente documentación).
- Una aceptación final del producto (a partir de la cual se dará por finalizado el proyecto).

El resultado de este planteamiento es, como se indica en [6]:

- La firma de requisitos o del análisis funcional crea un contexto que favorece que el cliente pida todo lo que se imagine que en algún momento quizás pueda ser necesario, ya que no dispondrá de ningún otro momento natural para realizar nuevas peticiones. Esta "carta a los reyes" favorece la aparición de requisitos no tan relevantes, en línea del informe de Standish Group [7]. Adicionalmente, obliga a realizar hipótesis antes de disponer de toda la información necesaria y de poder entender mejor el producto y contexto en que se va a desarrollar. Las hipótesis que no hayan sido correctas llevarán a la necesidad de arreglos (caros dado que los ajustes se hacen en fases tardías del proyecto) y, en el peor caso a productos mediocres (por la dificultad,

falta de tiempo y de presupuesto para rehacer partes del producto de manera adecuada).

- Durante una gran parte del proyecto la documentación intermedia “lo aguanta todo”, no se han atacado todavía los riesgos reales de la construcción real del producto.
- Se deja para la última etapa del proyecto una parte tan crítica como la verificación del cumplimiento de las expectativas del cliente respecto al producto final. Esto impide detectar a tiempo desalineamientos en requisitos que habría salido más barato corregir cuando todavía los tenían frescos las personas que los desarrollaron, sin haber construido más producto encima. Por otro lado, las hipótesis erróneas y los defectos todavía no encontrados (por no disponer de feedback) pueden multiplicar su efecto a lo largo del proyecto.

Las peores consecuencias de todo esto son que el cliente llegue a situaciones como:

- Pagar más de la cuenta y tener menos de lo esperado.
- Tener “todo o no nada” (si el proyecto pasa por una situación difícil que acabe en su cancelación).
- Retrasos del “todo”.

1.6. Agile es Gestión de Riesgos, Protege Más al Cliente

Los planteamientos ágiles (entre otras cosas) están enfocados a gestionar riesgos:

- Desarrollo de producto en bloques cortos, de manera que siempre hay producto estable, listo para ser puesto a disposición del usuario o consumidor final. Con ello también se obtiene más visibilidad del estado real del proyecto, de los objetivos ya conseguidos y de la velocidad de desarrollo.
- El cliente prioriza los objetivos del proyecto por el valor que le aportan y revisa producto tangible a lo largo de todo el proyecto con el fin de:
 - Asegurar desde los inicios del proyecto que el producto real ya desarrollado siempre esté alineado con sus expectativas (feedback anticipado).
 - Disponer de tiempo suficiente para realizar ajustes de los aspectos más críticos del producto.
 - Proporcionar tranquilidad a todos los implicados, ya que a la mitad del proyecto el “core” del producto ya está desarrollado, lo que queda es “ir añadiendo piezas”.
- De manera natural se proporciona flexibilidad a repriorizaciones, cambios e incluso a finalizar el proyecto de manera anticipada, si el resultado es suficientemente bueno o si no se desea continuar con el proyecto por alguna razón. Es decir, el cliente dirige los resultados del proyecto para poder satisfacer sus necesidades reales, con el apoyo del proveedor

Así pues, los planteamientos ágiles protegen más al cliente de cosas que puede que no se conozcan, ni él ni su proveedor.

2 Agile Explícito en el Contrato

Si se reconocen como adecuados los principios y prácticas ágiles para guiar la relación entre las partes, es conveniente explicitar estas reglas del juego en forma de cláusulas del contrato. Así encontraremos definidos aspectos como, por ejemplo:

- Priorización inicial por valor para el negocio de la lista de objetivos del proyecto o servicio (Product Backlog), que establecerá un primer punto de partida, pero no será vinculante.
- Revisiones regulares del producto final de modo que se disponga de ciclos cortos de aprendizaje, con posibilidad de repriorización del Product Backlog.
- “Change for free”, o cómo hacer cambios de requisitos a cuenta del esfuerzo pendiente en el proyecto.
- Finalización anticipada del proyecto.
- Definición de Hecho (*Definition of Done, DoD*), de manera que siempre haya producto preparado para ser utilizado.

3 Maneras de Cocinar Contratos

A continuación se muestran diferentes maneras de cocinar contratos ágiles así como posibles guarniciones (cláusulas adicionales que puede ser interesante incluir en el contrato en función del contexto). Estos contratos se pueden organizar en función de si las variables alcance, importe y plazos son fijas o variables (notar que en todas las modalidades de contratos se trabaja bajo principios Agile: iteraciones con producto estable y revisiones del cliente, equipos multidisciplinares autoorganizados que realizan retrospectivas, etc.).

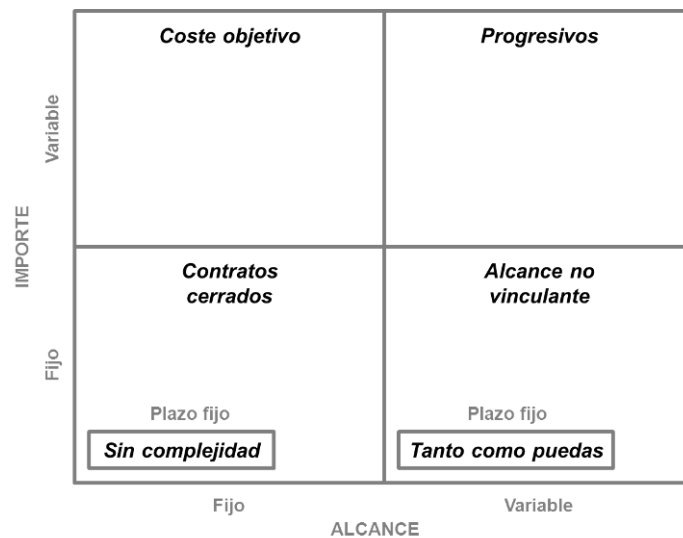


Fig. 4. Modalidades de contratos en función de las variables que se fijan.

Si consideramos que cada modalidad de contrato es un tipo de “cocina”, podríamos imaginar que existen las siguientes “cocinas de contratos”:

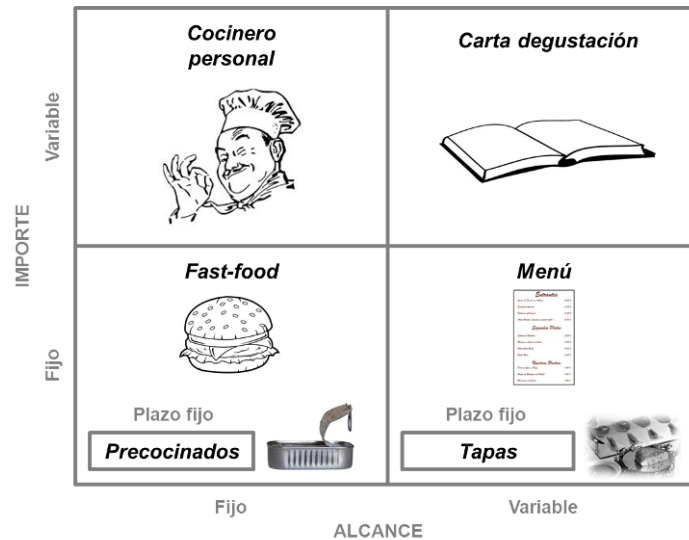


Fig. 5. Maneras de cocinar contratos.

3.1. Contratos “Fast-food” - “Cerrados” (Alcance Fijo e Importe Fijo)

Esta modalidad de contrato funciona bien en proyectos y servicios donde la complejidad e incertidumbre es baja, cuando el cliente sólo tiene opciones concretas a escoger y la calidad de servicio es conocida, por ejemplo en un servicio que se puede proporcionar de manera estándar (como en una hamburguesería) [8]. Estos servicios pueden ser altamente procedimentados (la inteligencia reside en el proceso) y no se producen situaciones donde las personas que proporcionan estos servicios tengan que tomar decisiones difíciles.

Si este no es el caso, es decir, se trata de un proyecto con indeterminación y complejidad alta (según la definición de complejidad descrita en el apartado 1.3) pero el cliente exige que se ejecute bajo un contrato de alcance fijo e importe fijo, una estrategia adecuada sería:

- No aceptar cambios.
- Contar con suficiente margen para:
 - Ajustes acotados (para poder incorporar un mínimo de feedback del cliente).
 - Imprevistos, tanto en el qué (entendimiento de los requisitos del producto) como en el cómo (estimaciones equivocadas, aspectos técnicos que se desconocen).
 - Cambios de priorizaciones.
 - Mejoras de la Definición de Hecho durante el proyecto.
- Configurar un equipo de expertos capaces de asegurar el éxito del proyecto.

Esta modalidad de contrato se suele aplicar cuando todavía no existe suficiente confianza con el proveedor. Un primer paso para que todas las partes minimicen riesgos, manteniendo el planteamiento de esta modalidad de contrato, es descomponer un proyecto en varias fases (e incluso iteraciones) de alcance fijo e importe fijo.

El escenario más complicado se da cuando, además de fijar alcance e importe, también se fijan los plazos. La ejecución de estos contratos requiere que el proyecto apenas tenga complejidad, que todo esté bajo control. Hasta a la cocina fast-food le puede resultar difícil cumplir si hay indeterminación o aparece algún imprevisto (incluida una cola repentina de clientes). Sería como plantearse comer algo concreto por un importe determinado y en un tiempo muy acotado, sin esperas ni riesgos, con una calidad media pero conocida, es decir, como recurrir a una nevera o congelador a por un precocinado.

3.2. Contrato “Menú” - “Alcance No Vinculante” (Alcance Variable e Importe Fijo)

En esta modalidad de contrato es conveniente explicitar la reemplazabilidad de objetivos del alcance. Por ejemplo, estableciendo un Product Backlog inicial no vinculante.

Sería el caso de un restaurante de menú de mediodías, donde el cliente se sienta sabiendo que, por un importe fijo, puede escoger entre un primer plato, un segundo plato y postre que puede cambiar durante la comida (el segundo plato o el postre), si avisa con suficiente antelación.

Guarniciones:

- **“Cambios gratis”** (*change for free* [3]). El cliente puede hacer cambios sobre los requisitos todavía no desarrollados, siempre que la reestimación del cómputo de horas restante no se incremente. Ver ejemplos de cláusula en [4].
- **Finalización anticipada.** El cliente puede cancelar el contrato en cualquier momento (por ejemplo si el resultado conseguido hasta el momento es suficientemente bueno – o malo).
- **“Dinero a cambio de nada”** (*money for nothing* [3]). Es una finalización anticipada en la que se protege también al proveedor, que posiblemente haya contado con los ingresos pendientes del proyecto durante unas fechas, para lo cual haya bloqueado a los miembros del equipo o rechazado otros contratos. El cliente paga un porcentaje del dinero restante al proveedor (por ejemplo un 20%, de manera que se ahorra un 80%). Ver ejemplos de cláusula en [4].
- **Pago de tareas adicionales.** Las tareas que se solicitan durante el proyecto y que no se identificaron en el contrato se pagan por esfuerzo dedicado – Por ejemplo: valoraciones o estudios extra, subidas a producción adicionales a las previstas, etc.

Si no se dispone de suficiente confianza con el cliente, una estrategia que el proveedor puede seguir es primero completar los requisitos *baseline* (los identificados en el contrato) antes de aceptar grandes cambios o requisitos nuevos [8].

El escenario en que, además de fijar el importe, también se fijan los plazos, puede ser aplicable en proyectos de consultoría como la elaboración de un Product Backlog o RFP que establezca una visión común de alcance, criterios de éxito y estimación de importe para el siguiente proyecto donde se hará el desarrollo. El Product Backlog se elabora en un tiempo máximo donde el alcance es variable en análisis y en contenido, en función de lo que sea más importante entender. Es decir, “haz tanto como puedas” en ese plazo. Sería como ir de tapas, con tiempo e importe limitados, en que sobre la marcha se va decidiendo qué tapas ir comiendo, en función de lo que apetezca más.

3.3. Contrato de “Cocinero Personal” - “Coste Objetivo” (Alcance Fijo e Importe Variable)

En el contrato de coste objetivo (*target cost*) se comparten tanto las ganancias (beneficios) como las pérdidas (costes). Es el modelo de trabajo que Toyota utiliza con sus proveedores. Su espíritu es que cliente y proveedor trabajen juntos para mejorar de manera continua, ser más productivos y reducir el margen para imprevistos del proveedor. Para ello necesita de transparencia en la contabilidad de los costes.

Supongamos el siguiente caso, basado en [2]:

Tabla 1. Ejemplo de pago del cliente en un contrato de coste objetivo.

Estimación del coste objetivo por parte del proveedor	Beneficio objetivo del proveedor	Pago objetivo del cliente
1000	150	1150

Tabla 2. Ejemplo de pago real del cliente si se produce una desviación en los costes. El cliente no los paga por completo, sólo un porcentaje de la desviación, en este caso del 60%.

Coste real	Beneficio para el proveedor	Pago del cliente
1100 (superior al estimado)	110 (inferior al previsto)	1210 (asume un 60% del sobrecoste)
900 (inferior al estimado)	190 (superior al estimado)	1090 (se queda con un 60% del ahorro)

En el pago de la desviación se incluirían:

- Arreglos del proveedor porque no entendió lo que se esperaba.
- Clarificaciones del cliente, necesita que se modifique algo bien desarrollado pero que no se supo o no se pudo explicar adecuadamente antes de su desarrollo.

No se considerarían desviaciones:

- Requisitos nuevos, que se deberían pagar en un 100%.
- Corregir defectos, que no se pagarían.

Guarniciones:

- **Porcentaje de pago por desviación revisable** cada N iteraciones.
- **Límite de pago por el cliente** en cada iteración / hito.
- **Límite de beneficio del proveedor** en cada iteración / hito si el importe excede una determinada cantidad.

Esta modalidad de contrato puede asimilarse al contrato regular de un cocinero personal al que se le pide que elabore diferentes cenas. El cocinero hace un presupuesto (que el cliente acepta) y va a comprar los ingredientes bajo su cuenta y riesgo, aunque el cliente paga un poco más si los ingredientes fueron más caros y un poco menos en caso contrario. En un caso ideal, el cliente asumiría todo el sobrecoste de los ingredientes (o se quedaría con todo el ahorro si el cocinero encuentra una oferta). En un ambiente empresarial, el pago compartido del porcentaje de desviación fomenta que las partes colaboren y vayan mejorando para que el resultado cada vez sea mejor y con menor indeterminación, especialmente si los costes tienen una parte debida a la relación entre ambas partes.

3.4. Contrato “Carta degustación” – “Progresivos” (Alcance Variable y Precio Variable)

Esta modalidad de contrato se basa en el pago por fase o iteración. Se suele utilizar, además de en proyectos, en servicios anuales. Se pueden encontrar diversos tipos de contratos en función de si el pago es más o menos fijo:

- **Pago fijo por fase o iteración.** Es una mejora a los contratos “cerrados” (explicada anteriormente), donde para gestionar mejor los riesgos los proyectos se dividen en fases y se acuerda el contenido de cada una. El pago fijo por iteración es sencillo de entender y de gestionar, resultando especialmente adecuado cuando hay un equipo de trabajo fijo en cada iteración o fase (como en Scrum o en servicios de *continuous release*).
- **Pago por coste de la fase o iteración (*Time & Materials*).** El cliente paga por el esfuerzo y recursos dedicados por el proveedor. Un planteamiento más sofisticado sería un pago por objetivos o productividad en forma de precio fijo por unidad de trabajo (Puntos de Casos de Uso, Puntos de Historia, Puntos Función, etc.). Para ello es necesario acordar qué es un punto y su precio, por ejemplo a través de la media de varios proyectos o realizando una media de la producción de varias iteraciones (lo cual necesita de un seguimiento detallado de costes).

Esta modalidad de contrato puede asimilarse a un restaurante de carta donde el cliente hace peticiones de platos que pueden ser muy diferentes unos de otros, requieren su elaboración particular y pueden tener coste diferente, hasta que ya no necesita encargar más. Notar que hay un abanico de cocinas que proporcionan este servicio, desde restaurantes chinos hasta con estrellas Michelin. El cliente escoge y compra en función de lo que necesita (así como un buen proveedor tiene una estrategia de producto, sabe a qué tipo de clientes quiere ofrecer sus servicios y puede escogerlos).

Guarniciones:

- **Límite de pago del cliente** en el proyecto o en cada iteración / hito / fase / *release* / año. Por ejemplo, se consume una bolsa de horas.
- **Si el coste de una fase es mayor de lo estimado, el cliente paga sólo una parte del precio** (semejante a los contratos de coste objetivo).
- Para proporcionar flexibilidad extra, **antes de iniciar una fase o iteración el cliente acuerda con el proveedor qué requisitos son “Must” y cuáles son “Should”**. De esta manera si durante el desarrollo aparecen necesidades nuevas que sea necesario incluir, ya se hizo el ejercicio de pensar qué requisitos se deberían desplazar a otra fase o iteración.

4 Framework de aceptación

En los proyectos y servicios ágiles la aceptación de producto se hace de manera continuada. Es importante que esta aceptación sea con suficiente calidad y dedicación por parte de todos los implicados, con el objeto de obtener un buen feedback y avanzar con paso seguro. A continuación se indican algunas reglas que puede ser interesante incluir en el contrato. En corchetes angulares (<< >>) figuran los aspectos más dependientes del tipo de proyecto o servicio:

- Cada requisito deben tener asociados sus criterios de aceptación antes de iniciar su desarrollo. Estos criterios serán la base para la revisión de cada requisito, ayudando a verificar si se ha desarrollado el producto esperado.
- La aceptación del producto tendrá lugar al final de <<cada desarrollo de requisito / iteración / hito / fase / *release* >>. Constará de <<2>> pasos:
 - Revisión presencial, en que el equipo mostrará el producto desarrollado al cliente y las personas que él determine como, por ejemplo, usuarios clave (los receptores finales del producto).
 - Período de aceptación. El cliente dispondrá de un plazo de <<N>> días tras la revisión para mostrar su conformidad con el producto desarrollado. Las disconformidades detectadas dentro de este plazo serán evaluadas para su corrección dentro de la <<iteración >> ya iniciada. En caso de superarse esta fecha, el proveedor podrá posponer su resolución a la siguiente <<iteración>>.
 - El cliente asistirá al proveedor en la corrección y revisión de no conformidades en el mínimo tiempo posible.
- Cada requisito debe ir acompañado de tests de regresión automatizados (que ayuden a verificar tanto que el requisito ha sido cubierto como que se sigue cubriendo en siguientes revisiones) en función de si se cumplen ciertos criterios de relevancia.
- Todos los requisitos deben cumplir con la siguiente Definición de Hecho (*DoD*), que establece los entregables a elaborar y acciones a realizar, de

manera iterativa e incremental, para asegurar que en todo momento el producto se encuentra en situación de ser utilizado, con un mínimo esfuerzo de puesta en producción final <<insertar aquí la Definición de Hecho>>. En caso de ser necesaria una ampliación de la *DoD*, se renegociaría con el proveedor el impacto en las estimaciones.

Uno de los riesgos de proporcionar flexibilidad a ajustes y nuevos requisitos al final de cada iteración, hito o fase es que el cliente pierda la perspectiva de la globalidad del proyecto y que en su finalización reclame gran parte del alcance inicial comprometido, al darse cuenta de que no ha progresado suficientemente (por haber dirigido los esfuerzos a cambios menores y arreglos, que él mismo solicitó). Es por ello que puede ser interesante mostrar, al final de cada iteración:

- Una visión del *baseline*, el trabajo ya realizado, el pendiente y los cambios o añadidos respecto al alcance global, en forma de mapa de producto.
- La velocidad de desarrollo y la proyección de fecha estimada de finalización.

4 Conclusiones

Aunque el cliente imponga un contrato cerrado (que sólo funcionaría bien para hacer “fast-food”) sigue aportando valor el aplicar los principios y prácticas ágiles: iteraciones con producto estable (permiten saber con claridad y con tiempo en qué punto se encuentra el proyecto y tomar decisiones), con revisiones del cliente a ser posible, equipos multidisciplinares autoorganizados (capaces por sí mismos de coger un requisito, desarrollarlo por completo, probarlo y llevarlo a producción) que realizan retrospectivas regulares (para mejorar continuamente), etc.

Si el cliente solicita trabajar con un contrato cerrado en alcance y coste, una estrategia para ayudar a que el proyecto cada vez sea más productivo es dividirlo en fases, generar confianza e ir negociando cada vez contratos más ágiles para las siguientes fases: ir introduciendo contratos de precio y plazo fijo (alcance variable), progresivos (por ejemplo *Time & Materials*) y de Coste objetivo.

Referencias

1. Agile Manifiesto - <http://agilemanifesto.org/>
2. Arbogast, T., Larman, C., Vodde, B. – Agile Contracts Primer - <http://www.agilecontracts.org/>
3. Sutherland, J. - Money for Nothing and Your Change for Free - <http://jeffsutherland.com/Agile2008MoneyforNothing.pdf>
4. Albaladejo, X. – Un contrato ágil para Scrum, <http://www.proyectosagiles.org/contrato-agil-scrum>
5. Beaumont, S. – Agile & Contracts, www.scrumalliance.org/resource_download/442
6. agile-spain.org, ProyectosAgiles.org - La alternativa ágil - <http://www.slideshare.net/xalbaladejo/la-alternativa-agil-v57>
7. Johnson, J. - The Standish Group International Inc - <http://martinfowler.com/articles/xp2002.html>
8. Medinilla, A. - Contratos ágiles - <http://www.slideshare.net/proyectalis/110115-contratos-agiles>