

Metodologías Ágiles: La ventaja competitiva de estar preparado para tomar decisiones lo más tarde posible y cambiarlas en cualquier momento

Jesús Pérez Sánchez
jesús.perez@germinus.com
Agile-Spain / Germinus

Resumen

Tomar las decisiones cuando no se está en condiciones de llevarlas a cabo es probablemente uno de los mayores problemas a los que se enfrentan aquellas empresas que contratan el desarrollo de un proyecto para Internet. Los proyectos suelen ser largos y suelen definirse al principio de la contratación, dejando poco margen para el cambio en un entorno muy cambiante como el de Internet.

El entorno de Internet es un entorno que se mueve, donde los avances tecnológicos tanto en software como en infraestructura, así como las nuevas necesidades, hacen que los escenarios cambien en pocos meses. En estos entornos es imprescindible estar preparado para que el objetivo de nuestros desarrollos cambie, porque estarlo será una ventaja competitiva fundamental.

Metodologías Tradicionales

El concepto de Ingeniería del Software llegó después de producirse la crisis del software. El desarrollo de software era excesivamente artesanal y no permitía planificar y estimar el esfuerzo de una manera razonable. Los proyectos eran muy ambiciosos y la ausencia de metodologías en muchas ocasiones acababa en un caos. Por este motivo se importaron metodologías de otros campos donde también existían procesos de Ingeniería. Se trataba de procedimentar y documentar todo el proceso, para minimizar el riesgo de cada proyecto y controlar su evolución. Estas metodologías importadas son las que conocemos como metodologías tradicionales.

Las metodologías tradicionales abordan estos problemas proponiendo comenzar con una fase de análisis, en la que se tomen todas las decisiones, previa al comienzo del desarrollo. Finalizada esta fase de análisis será el momento comenzar el desarrollo que debe finalizar con una etapa de pruebas que asegure la calidad antes de implantar el sistema en producción. Para llevar a cabo todo el proceso proponen una serie

de documentos que se deben realizar a lo largo del proyecto. Esta documentación que mayoritariamente debe realizarse en la primera fase, nos permitirá tener un plan de proyecto y entender todas las decisiones que se aplican en cada momento del proyecto.

Las mayoría de las metodologías tradicionales definen un proceso secuencial donde cada proceso se alimenta del anterior y en el que el software esta disponible al final de todo el proceso. Son los llamados procesos en cascada [Ref.5].

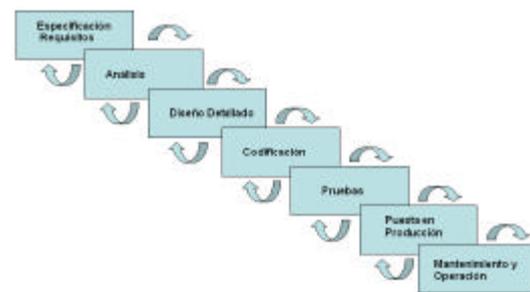


Figura 1. Procesos en Cascada 1

La adopción de estas metodologías fue una sobre-reacción ante la crisis del Software. Esta aproximación era una respuesta de naturaleza defensiva ante los problemas que se habían detectado en el desarrollo de software. Estos problemas se encontraban en la incorrecta estimación del proyecto y en la compleja ejecución técnica del proyecto. Estas metodologías nos proponen solucionarlos definiendo correctamente el alcance de los proyectos y resolviendo las dificultades técnicas antes de comenzar la ejecución de un proyecto. Para definir el alcance nos proponen una detallada especificación de los requisitos para eliminar en lo posible su ambigüedad. Para eliminar las dificultades técnicas que podamos encontrarnos en el entorno tecnológico volátil en el que se suelen desarrollar los proyectos, proponen una primera fase donde se detalle la solución técnica que debe ejecutarse, tratando de

despejar cualquier duda tecnológica antes de comenzar.

Estos problemas se hicieron evidentes debido a la crisis del software donde la estimación de esfuerzo era claramente insuficiente para los ambiciosos alcances que se definían y que a la postre era generalmente la razón del fracaso del proyecto.

La estimación de un proyecto de desarrollo tiene tres principalmente tres variables: el tiempo, el coste y los requisitos. Podríamos incluir la calidad como la cuarta, pero dado lo complicado que es de especificar no suele utilizarse en la estimación. En una oferta de servicios profesionales las variables de tiempo y de coste se especifican sin ningún tipo de ambigüedad, mientras que el alcance es muy difícil de detallar y su interpretación en muchos casos es subjetiva. Este es uno de los principales problemas que intentan resolver las metodologías tradicionales intentando eliminar toda la ambigüedad al definir el alcance. Cuanto mayor sea el proyecto, mayor será el esfuerzo en definir el alcance de una forma clara y precisa. Se trata de conseguir delimitar lo que está contratado de lo que no está. Una definición que deberá en muchos casos entrar en mucho detalle para ser realmente útil. Una vez definida esa delgada línea, servirá de referencia en todo el proyecto para defendernos de nuevos requisitos.

Definir el alcance del proyecto no resuelve complementemente el problema de estimar el esfuerzo que será necesario para llevar a cabo los objetivos definidos. Si conseguimos cerrar un alcance pero no somos capaces de estimar el esfuerzo necesario para llevarlo a cabo no habremos conseguido reducir el riesgo que tratábamos de minimizar. La predicción del esfuerzo necesario sólo es posible basarla en la experiencia pasada acometiendo proyectos de similar complejidad y analizando el alcance de las tareas que debemos realizar. Para llevar a cabo una estimación necesitaremos profundizar en esa especificación inicial para abordar detalles de tipo técnico que nos permitan razonablemente conocer el esfuerzo relacionado con cada tarea a realizar. No vale con conocer los requisitos, tendremos que conocer con que herramientas contaremos, cuando tendremos disponibles otros sistemas con los que tengamos que integrar, etc. Este análisis nos permitirá tener un plan detallado que dirigirá nuestro proyecto y un detalle sobre como se abordará técnicamente.

Antes de comenzar a desarrollar o de arrancar nuestro proyecto deberemos tener la inmensa mayoría de los detalles cerrados. Las metodologías tradicionales justifican este planteamiento porque con toda esta información se podrá encontrar la mejor solución técnica para el problema que se plantea y se podrán planificar el uso de recursos más óptimo. Tendremos toda la información para ser lo más eficientes en la ejecución del proyecto dado que tenemos todos los detalles de lo que se quiere hacer.

Nuestra planificación nos dará un mapa exacto de cuál será nuestro viaje. Tendremos todos los detalles del camino que tendremos que recorrer. Siguiendo nuestra metodología no arrancaremos nuestro vehículo hasta que hayamos analizado la ruta, elegido el vehículo que mejor se adapte al recorrido e incluso programado nuestro vehículo para que adapte la conducción a las curvas que tendremos que tomar. Pero lo que no sabemos es cómo se comportaría en otros recorridos. No tendremos la posibilidad de cambiar el recorrido a mitad de viaje porque eso nos obligaría a reprogramar nuestro vehículo e incluso a cambiarlo, pero si nuestro recorrido no cambia seremos probablemente de los más rápidos.

¿Cuáles son los problemas?

La experiencia nos ha demostrado que las metodologías tradicionales no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, porque no están pensadas para trabajar con incertidumbre. No están preparadas para el cambio.

Aplicar metodologías tradicionales nos obliga a forzar a nuestro cliente a que tome todas las decisiones al principio. El verdadero problema no será el forzarnos a detallar este alcance con él. Detallarlo siempre ayudará a conocer mejor los requisitos y a que el cliente entienda mejor sus necesidades. El problema será que estos detalles nos harán tomar decisiones que luego serán muy costosas de cambiar.

Nuestro cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él. Este alcance dará lugar a muchas decisiones técnicas que serán muy costosas cambiar y de las cuales en alguna ocasión nuestro cliente no será consciente de las implicaciones que tienen. Estamos poniendo a nuestro cliente en una situación que puede ser muy incómoda para él.

Debe decidir el alcance de un proyecto que puede durar más de un año, en una primera fase que puede durar unos meses y donde no se avanzará nada en las necesidades a corto plazo que tiene.

La situación es más incómoda cuanto mayor grado de incertidumbre tenga sobre el alcance del proyecto. Esto también estará relacionado con el tamaño del proyecto. La incertidumbre será mayor cuanto mayor sea éste. Es parecido a planificar un viaje. Si tuviéramos que planificar un viaje de una semana podríamos entrar en mucho detalle, pero si queremos realizar un viaje de un año ¿seríamos capaces de reservar todos los hoteles y vuelos de avión? ¿No agradeceríamos poder hacer esta reserva conforme avanzara nuestro viaje? Probablemente valoraríamos esta flexibilidad en esos casos, como por ejemplo lo valoramos en otras situaciones. Cuando compramos un regalo llevamos cabo una decisión que tiene mucha incertidumbre dado que es complicado acertar, por este motivo valoramos que nuestro amigo pueda posteriormente cambiarlo aunque pueda costarnos algo más. Tomar decisiones irrevocables en entornos de incertidumbre es una situación incómoda.

En ámbitos financieros existen instrumentos como son las opciones que nos permiten, mediante una cantidad mucho menor (primas), tener la posibilidad de llevar a cabo una inversión en acciones en un futuro. Pero es una decisión que podremos realizar o no dependiendo de lo que nos interese cuando llegue esa fecha sin tener que llevar a cabo la inversión total a priori. Son instrumentos para manejar la incertidumbre. En el desarrollo de software también podemos utilizar estas estrategias. Si podemos dejar abiertas varias posibilidades, podremos elegir más tarde y con más información cual será la mejor elección (incluso podría ser no hacerla) y esto a cambio de realizar una inversión mucho menor que es permitir esa flexibilidad.

Ante el problema de tener que detallar todo prematuramente, el cliente suele adoptar una posición defensiva que es contraria a sus intereses. El cliente tratará que el sistema sea lo suficientemente flexible para poder cubrir las necesidades que pudieran surgir o cambiar. Por este motivo el sistema implementará ciertas funcionalidades no se llegaran a utilizar. El cliente no es consciente de que cuando toma estas decisiones está introduciendo una complejidad en el proyecto que reducirá el

alcance de otras funcionalidades y hará que el sistema sea más complejo de gestionar y de extender. Limitará el alcance de su proyecto final porque para el mismo esfuerzo de desarrollo tenderá a tener menos funcionalidades de valor para él. Quizás una negociación hábil pueda conseguir que la empresa de servicios se vea obligada a realizar mayor esfuerzo del que estimó, pero sin duda esto también será un problema para él. Cuando una empresa de servicios tiene un proyecto que se convierte en deficitario las decisiones técnicas que se empiezan a tomar suelen ir guiadas más por la reducción del coste, que por la excelencia técnica, con lo cual al final lo verá reflejado en la calidad del software que se le entrega. La empresa de servicio intenta minimizar ese sobre-esfuerzo en la variable de la estimación que menos visible es: la calidad.

Para tomar estas decisiones el cliente tiene como única realimentación la documentación técnica que se va generando en la primera fase de diseño. Dado que esto es claramente insuficiente en muchos casos se ayuda al cliente con prototipos para llevar a cabo estas decisiones. Pero esto tampoco será realmente útil, puesto que el usuario final no suelen ser las personas del cliente responsables de definir el alcance. Muchos de las necesidades del software se encontrarán cuando el software ponga en producción y comience a ser utilizado por sus usuarios finales.

Las decisiones que vaya tomando el cliente serán irrevocables o muy costosas. Pero ¿por qué son estas decisiones tan costosas de cambiar? ¿Por qué ciertas decisiones no pueden ser tomadas conforme avanza el proyecto? ¿Cuál es el objetivo del proyecto? ¿Acabar en tiempo y hacerlo técnicamente perfecto, o maximizar el valor que aporta al cliente? ¿En que caso es más probable que el cliente siga contando con la empresa de servicios: cuándo el proyecto ha tenido éxito para el cliente o cuando se ha realizado técnicamente de forma perfecta?. Quizás haya que ponderar la satisfacción del cliente tanto como la excelencia técnica.

Coste del Cambio.

Las metodologías tradicionales como hemos hablado definen un proceso secuencial en el que se realiza el desarrollo de software. Dado que cada proceso es alimentado por el anterior el coste de un cambio es mayor cuanto más tarde se produce dado que es necesario volver a repetir las tareas que ya se habían realizado. La curva

del coste aplicando metodologías tradicionales [Ref. 3] es:

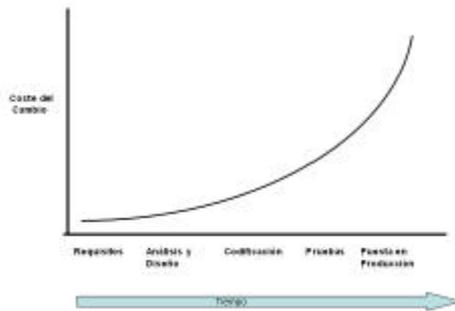


Figura 2 .Coste del Cambio en M. Tradicionales 1

El coste del cambio está muy relacionado con la metodología que se elige para llevar a cabo el proceso. Una de las características negativas de las metodologías tradicionales es que inducen que éste sea muy elevado.

Pero también existen otras variables que afectan al coste del cambio. Una de ellas es la complejidad. Cuanto más complejo es un proyecto, mayor será el coste del cambio. Si queremos reducir el coste del cambio deberemos tratar de que nuestro sistema sea lo más sencillo posible.

Si se ha desarrollado el proyecto teniendo en cuenta que será necesario cambiar ciertas funcionalidades, los cambios serán menos costosos. Si los diseñadores de los coches únicamente hubieran pensado en el diseño del coche más rápido, probablemente hubieran hecho una carrocería que lo cubriera totalmente, pero ¿Cuál sería el coste de cambiar una rueda?. Permitir cambiar una rueda al coche en software es similar a desarrollar código con bajo acoplamiento y alta cohesión.

Dado que los cambios son muy costosos cuando aplicamos metodologías tradicionales es necesario avisar al cliente de que el cambio de las decisiones que está tomando en la primera fase tendrán un coste elevado conforme el tiempo vaya pasando.

Se puede intentar reducir el coste del cambio aplicando las técnicas que anteriormente comentábamos pero también es importante tener en cuenta lo que las metodologías ágiles han puesto de relevancia: “Una decisión que no ha sido tomada es evidentemente mucho más fácil de cambiar”. Retrasar las decisiones es una de las

estrategias que permiten reducir el coste del cambio.

Metodologías Ágiles

Como respuesta a los problemas aplicando metodologías tradicionales surgieron otras metodologías que trataban de adaptarse a la realidad del desarrollo de software. En 2001 los principales promotores de estas teorías se reunieron para debatir los puntos que tenía en común y el resultado fue el Manifiesto Ágil. Este manifiesto establece las ideas principales de las metodologías ágiles:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Hay diversos métodos ágiles que recogen estas ideas como: eXtreme Programming (XP),Crystal Methods ,SCRUM ,etc.

Estas metodologías ponen de relevancia que la capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significar reducir su coste.

Retrasar las decisiones

Las metodologías ágiles nos llevan a retrasar las decisiones tanto como sea posible (de una manera responsable), porque será una ventaja tanto para la empresa de desarrollo como para el cliente y no hay que olvidar que el éxito de una empresa desarrollo esta muy relacionado con el éxito de sus clientes.

Tomar decisiones es similar a invertir nuestro dinero, si lo invertimos todo no podremos aprovechar otras oportunidades de inversión que surjan. Y si lo invertimos queremos que este nos produzca beneficio desde el momento en que hacemos la inversión. Deberíamos tratar el desarrollo de software de la

misma manera, tratando de sacar la mayor rentabilidad a nuestros desarrollos o infraestructuras.

Las principales ventajas de retrasar las decisiones son

- Reduce el número de decisiones de alta inversión que se toman.
- Llevando a cabo una primera aproximación sobre estas decisiones, esto nos permitirá que estas se tomen con mayor seguridad
- Reduce el número de cambios necesario en el proyecto.
- Reduce el coste del cambio

El entorno tecnológico en el que desarrollan los proyectos de software avanza de una forma muy rápida e influye directamente en el proyecto. Es especialmente evidente con los elementos de Hardware cuya capacidad y eficiencia aumenta continuamente, dejando obsoletos rápidamente los ya existente. Si fuéramos a comprar Hardware cual sería la estrategia mas eficiente ¿No sería comprarlo justo en el momento que lo vamos a empezar a utilizar? En muchos proyectos junto con el desarrollo de software suele acompañarse de una oferta de maquinas. ¿Cuándo el proyecto va a durar un año, tiene sentido hacer esa compra al principio? En la mayoría de los casos será mucho mas eficiente hacerlo lo más tarde posible.

Retrasar las decisiones no significa no tomar decisiones, si no tomar las que sean necesarias en cada momento. No es razonable pensar que es posible dejarlo todo para el último momento como tampoco pensamos que sea razonable tomar todas las decisiones al principio. La toma de decisiones es un proceso que debe acompañar a todo el proyecto tomando las decisiones en los momentos adecuados que para la mayoría de las decisiones será el último momento en el que de manera responsable se puede tomar llevar a cabo la ejecución de decisión de una manera organizada. Dejar muy poco tiempo para ejecutar las decisiones aumentaría el riesgo del proyecto.

Existen decisiones que es necesario tomarlas al comienzo del proyecto. No todas las decisiones son del mismo tipo y por este motivo es necesario analizar que decisiones pueden ser retrasadas. Si analizamos en detalle número de estas decisiones que no podemos retrasar no daremos cuenta que no es un numero tan alto y

que probablemente en nuestro proyectos estemos tomando muchas decisiones que pudiéramos retrasar. Quizás la elección del lenguaje de programación o utilizar algún producto ya desarrollado puedan ser un ejemplo de decisiones que es necesario tomar al principio del proyecto.

Planificación Predictiva Vs Planificación Adaptativa

La aproximación de las metodologías ágiles es de naturaleza agresiva. Como se pone de relevancia en el manifiesto se trata de maximizar el valor del software que se le entrega al cliente por encima de tener un plan detallado. Es de naturaleza agresiva porque nos propone potenciar la relación con nuestro cliente en lugar de protegernos contra el. Nos propone comenzar a desarrollar cuando tengamos detallado el corto plazo, sin esa necesidad de tener todo decidido, nos propone desarrollar teniendo en cuenta que se llevarán a cabo cambios en medio del proyecto. Estamos siendo agresivos porque queremos maximizar el valor que entregamos al cliente y asumimos trabajar con cierta incertidumbre para conseguir ese objetivo.

Esto implicará que planificaremos para desarrollar en cada momento lo que más valor tenga para el cliente. Con un plan detallado probablemente seríamos más eficientes, aunque esa eficiencia la perderíamos en el momento que tuviéramos que cambiar alguna decisión costosa. Al final en proyectos donde existe mucha incertidumbre esta aproximación será probablemente mucho más eficaz para conseguir el objetivo.

Hacer una planificación adaptativa consiste en ir tomando las decisiones a lo largo del proyecto. Entraremos en detalle de especificaciones del desarrollo que se vaya a llevar a cabo en las próximas semanas a lo sumo. Entramos en detalle de lo que vayamos a desarrollar a corto plazo. Estamos transformando el proyecto en un conjunto de proyectos pequeños.

Las planificaciones a corto plazo nos permitirán tener software disponible para nuestro cliente y nos permitirán aprender de las estimaciones que vayamos realizando permitiendo que nuestra planificación sea sensible si nos encontramos con dificultades técnicas que ralentizan nuestro desarrollo o si por el contrario lo aceleran.

Para no perder la visión global será necesario tener una especificación de alto nivel del alcance del proyecto.

Planificar de manera adaptativa nos impedirá ver el futuro con claridad, dado que lo definiremos de una manera general, pero la pregunta quizás sería ¿No nos estamos engañando tratando de planificar en detalle a largo plazo? ¿No es condenarnos a fallar en la planificación?

También nos encontraremos con otras dificultades aplicando metodologías ágiles. El principal problema que nos encontramos cuando adoptamos planificaciones adaptativas es cómo delimitar el alcance del proyecto con nuestro cliente. La solución es definir un alcance a alto nivel en un primer momento basado en la experiencia que tenemos en el desarrollo de software. Esta predicción tratará de definir que funcionalidades a grandes rasgos se desarrollarán en el proyecto. Si los requisitos del proyecto cambiaran, sería necesario intercambiar algunas de las funcionalidades que se hayan estimado por las nuevas que se acaban de identificar.

Es una filosofía parecida a la subcontratación por horas, salvo que en este caso en lugar de horas tendremos funcionalidades. De esta manera habremos acordado realizar una serie de funcionalidades que tendrán cada una, una complejidad asociada, que podremos intercambiar por otras de similar complejidad. En cada iteración de nuestra planificación desarrollaremos una serie de funcionalidades que deberán ser entregadas al cliente como finales. De esta manera en cada iteración seremos capaces de plantear el objetivo de la siguiente y adaptarnos a las necesidades del cliente si estas cambiaran. El cambio es una situación normal dentro del proceso que definen las metodologías ágiles.

Este tipo de planificación es mucho más transparente para nuestros clientes. Esto supone una ventaja para ellos, ya que podrán conocer con mayor seguridad las fechas en las que estarán disponibles las diferentes funcionalidades de nuestro desarrollo. Esta necesidad por conocer las fechas en la que dispondrá del software en muchos casos es de vital importancia para su negocio. En una planificación adaptativa el cliente siempre tendrá el producto disponible con las funcionalidades que se hayan implementado hasta la fecha permitiéndonos poner en producción cuando se decida. Incluso nos permitirá definir en cada

iteración cuales son los objetivos de la siguiente, que se pueden orientar en cualquier momento para que el producto tenga un conjunto de funcionalidades coherentes para ponerlo en producción.

Poner en producción el producto de manera iterativo es una práctica que en nuestra experiencia reduce el riesgo del proyecto y permite tener realimentación de los usuarios muy útil. Esta realimentación permite que nuestro cliente pueda conocer las posibles mejoras de las funcionalidades para el software tenga más valor para el usuario final.

También tendrá un efecto positivo para el equipo de desarrollo dado que el efecto de la presión de las entregas se distribuirá a lo largo de todo el proyecto y no en una única entrega final. Decididamente no es una buena idea en los proyectos que solo se disponga de una entrega porque la presión y los retrasos se concentran en ella y esto suele acabar en una fase final de desarrollo con prisas que en ocasiones conducen a un caos difícil de recuperar.

Una de las ventajas más importantes de adoptar una planificación adaptativa es que estaremos preparados para el cambio, porque lo hemos introducido en nuestro proceso de desarrollo.

Planificar de manera adaptativa supone tomar las decisiones lo más tarde posible que es realmente lo que permite ser más ágiles. Lo que permite ir adaptando nuestro desarrollo hacia lo que es de mayor valor para nuestro cliente. En la siguiente gráfica podemos encontrar diferentes metodologías clasificadas en función de su adaptabilidad o agilidad



Figura 3. Decisiones y Agilidad 1

Técnicas para retrasar las decisiones

Para poder estar preparados para el cambio, también lo tiene que estar nuestro equipo de desarrollo. Estar preparados para el cambio requiere aplicar técnicas que lo faciliten.

Las diferentes metodologías ágiles proponen diferentes técnicas que permiten minimizar el coste del cambio. Tomar decisiones de una manera continua nos lleva a que nuestro diseño este en continua evolución. Técnicas como la refactorización permiten que nuestro diseño vaya adaptándose a las nuevas necesidades. Refactorizar significa modificar la estructura interna de nuestro código sin cambiar su comportamiento externo. Esta técnica permitirá adaptar nuestro diseño para que los cambios sean más fáciles de llevar a cabo. Y también permite que podamos extender nuestro código de la forma más eficiente.

Por otra parte, cuando aplicamos un cambio será necesario comprobar que no hemos creado efectos colaterales que hagan fallar nuestro desarrollo en las funcionalidades que no queríamos modificar. Aplicando pruebas unitarias sobre nuestro código podremos cambiar nuestro desarrollo con una mayor seguridad.

El coste de los cambios será proporcional a la complejidad de nuestro software. Intentando mantener nuestro diseño lo mas sencillo posible y lo menos acoplado posible permitirá que los cambios sean menos traumáticos.

Involucrar al cliente en el desarrollo nos permitirá que las decisiones puedan ser tomadas mientras se desarrolla el software y que los cambios por especificaciones incorrectas sean menores.

Estas prácticas anteriormente comentadas son parte del conjunto de prácticas que proponen las metodologías ágiles y que hará mucho más fácil retrasar las decisiones.

Experiencia en proyectos Reales

En proyectos reales la aplicación de las metodologías ágiles se enfrenta en muchas situaciones a problemas que la teoría no contemplaba.

El principal problema que nos encontramos en los proyectos reales es que los tipos de contratos no encajan con algunas de las prácticas. Estos contratos suelen ser de dos tipos:

- Contrato de Alcance Cerrado: Que nos obligan a cerrar los requisitos en la primera fase (Metodologías tradicionales)

- Subcontratación de personal: Que no podemos clasificarlo como proyecto, ya que suele ser el cliente el que dicta en que se emplea ese esfuerzo.

Una solución que hemos utilizado en proyectos ha sido definir un nuevo tipo de contrato, en el que especifiquemos la funcionalidad que se va a desarrollar, pero en el que el alcance podrá ser modificado durante la duración del contrato. Este alcance podrá ser modificado porque lo que se ofrece es un esfuerzo que es el que se estima que van a llevar los requisitos acordados, pero que podrán ser intercambiados por otros requisitos de igual dificultad técnica.

Las dificultades del contrato residen en la dificultad del cliente a aceptar un contrato donde no se detalle de una manera precisa el alcance del proyecto. Una de las soluciones es que el contrato se divida en una serie de entregas en las que se irán entregando funcionalidades ya finalizadas al cliente. En ese momento el cliente podrá optar por seguir con la siguiente entrega, cambiar su alcance o finalizarlo. Estamos dando al cliente la opción de que pueda retrasar las decisiones a cambio de no cerrar en detalle el alcance. La empresa de desarrollo tendrá mucho interés en que el cliente haya quedado satisfecho para continuar prestando servicios por lo que tratará de ofrecer el mayor rendimiento posible en cada una de las entregas. Kent Beck, una de las personas mas representativas de las metodologías ágiles, introduce el Contrato de Alcance Opcional que trata de formalizar esta idea.

Retrasar las decisiones en lo posible es también otra de las prácticas que mas difícil resulta aplicar. Los responsables de las empresas suelen preferir que todo se decida lo antes posible porque representa menos riesgo desde un punto de vista de gestión, pero en la mayoría de las ocasiones más ineficiente.

No obstante esta idea de retrasar decisiones no es nada nueva y en otros sectores como el de la logística o el financiero estas ideas ya existen y se aplican con éxito. Las empresas que no las aplicaron al final no han tenido más remedio que hacerlo.

Nuestra experiencia en los proyectos donde hemos conseguido que el cliente retrasara las decisiones en lo posible ha permitido que nuestro cliente tomara mejor las decisiones y que

incluso al final no hiciera inversiones que no le eran rentables.

Conclusión

La adopción de metodologías ágiles, y en concreto retrasar las decisiones, supone una ventaja competitiva tanto para el cliente como para la empresa de desarrollo, en proyectos donde exista mucha incertidumbre.

Llevando a cabo una planificación adaptativa permitimos que nuestro cliente tenga la flexibilidad de cambiar el alcance si lo cree necesario sin un gran coste. El cliente no se verá obligado a tomar decisiones prematuramente que conlleven importantes inversiones en tecnología que puede que nunca llegue a utilizar. El objetivo es que al final del proyecto el cliente tenga el software que mayor valor le ofrezca para la inversión que realizó.

Para la empresa de servicios profesionales las metodologías ágiles nos ofrecen una manera razonable de gestionar la incertidumbre en la estimación de los proyectos permitiéndonos ofrecer el mejor servicio posible a nuestro cliente.

Referencias:

1 Lean Software Development: An Agile Toolkit. Mary Poppendieck, Tom Poppendieck. Addison-Wesley

2 Agile Methodologies in Software Development for a Fast Moving World
Giancarlo Succi

3. Examining the Cost of Change
Scott W. Ambler

4 Extreme programming explained
Kent Beck. Addison-Wesley

5. A guide to Successful Software Development. Murray Cantor. Addison-Wesley, NY, 2002,

Jesús Pérez Sánchez
Agile-Spain / Germinus
jesus.perez@germinus.com
<http://www.germinus.com>