

Metodologías ágiles

mitos realidades

LAS METODOLOGÍAS ÁGILES, PRÁCTICAS OPUESTAS A LOS PROCESOS RÍGIDOS, Y APLICADAS DESDE HACE CASI DIEZ AÑOS EN EL MUNDO, TODAVÍA GENERAN RESISTENCIA BASADA EN MITOS QUE AQUÍ TRATAMOS DE DERRIBAR. UN ANÁLISIS EN TRES PARTES, QUE INCLUYE SUGERENCIAS Y CONSEJOS PARA IMPLEMENTARLAS DE MANERA EFECTIVA Y NO COMETER LOS ERRORES TÍPICOS. LA MIRADA DE LOS QUE YA LAS INSTRUMENTARON EN FORMA EXITOSA.

El término “ágil”(de “agilizar”), que la Real Academia Española define como “dar rapidez y facilidad al desarrollo de un proceso o a la realización de algo”, fue elegido en 2001 por un grupo de pensadores disidentes del mundo del software para calificar sus metodologías. A través del denominado –y ahora mítico– Manifiesto Agile, declaran valorar más los individuos e interacciones por sobre los procesos y herramientas. Además, priorizan el software que funciona, más que la documentación exhaustiva; aprecian la colaboración con el cliente, más que la negociación de contratos; y rescatan la respuesta ante el cambio, antes que el seguimiento de un plan. Casi una década después, estas metodologías “ágiles” han sido aplicadas en miles de proyectos de software en todo el mundo.

Extreme Programming (XP), Scrum, Lean y Kanban son probablemente las metodologías ágiles que más se usaron en la última década, pero siguen prevaleciendo mitos y malas interpretaciones que dificultan su adopción exitosa.

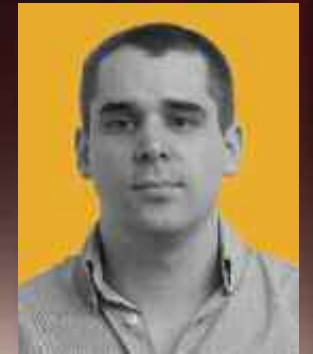
Los tres actos de esta nota intentan refutar los principales prejuicios que suelen aparecer a la hora de emprender el camino hacia la agilidad.

Primer acto

El primero de los mitos con el que uno se encuentra cuando comienza a hablar de metodologías ágiles es éste: “Es demasiado nuevo y son pocos los que lo usan”. Lo cierto es que a pesar de que estas metodologías se han puesto de moda hace poco, especialmente en la Argentina, tienen varios años y sus procesos han sido probados repetidamente. Repasemos algunas fechas importantes:

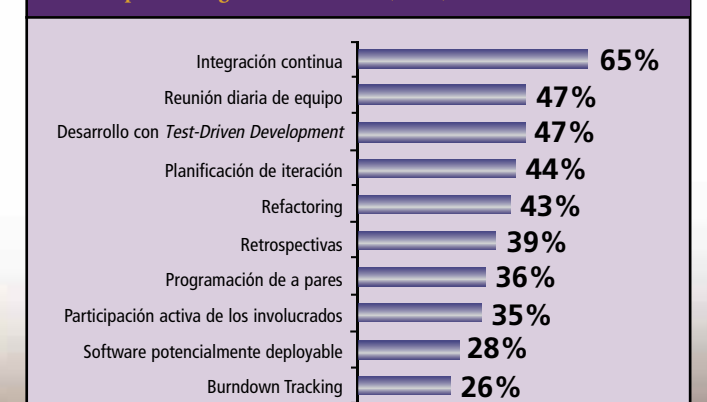
- La piedra fundamental de Scrum fue publicada hace 23 años por Hirotaka Takeuchi e Ikujiro Nonaka, y el primer proceso de este tipo fue concebido, ejecutado y documentado por Jeff

POR **THOMAS WALLET** PRAGMA CONSULTORES (ARGENTINA)



- Sutherland, John Scumniotales y Jeff McKenna hace 16 años.
- Extreme Programming fue concebido y aplicado por Kent Beck, Ron Jeffries y Ward Cunningham en el proyecto Chrysler Comprehensive Compensation System, hace 13 años. El método test-first development fue usado hace 50 años en el proyecto Mercury de la NASA.
- Los principios de Lean, llevados al desarrollo de software por Tom y Mary Poppendieck, fueron publicados hace sólo seis años. El concepto de *Just-In-Time*, uno de los pilares de Lean Manufacturing, fue introducido por Toyota en la industria automotriz hace 75 años.
- Kanban tiene sus orígenes en los sistemas mercantiles japoneses de

1- Las 10 prácticas ágiles más efectivas (de 30)



Fuente: “Agile Practices Survey 2009” - Scott W. Amber - Ambysoft

Enfoque en el negocio

POR HUGO A. CORONATO, GERENTE DE PROYECTOS, INENCO S.R.L., NEUQUÉN, ARGENTINA

“Aplicamos Scrum en el Proyecto Sistema de Administración Financiera de la provincia de Neuquén (Safipro), iniciativa que desarrolla la Contaduría General del Gobierno. Lo implementamos porque estamos convencidos de que en la naturaleza del proyecto se encuentran varios de los puntos que justifican la utilización de la metodología. Algunos de ellos son: la presencia de un entorno complejo donde se necesita obtener rápidos resultados, la imposición de requisitos cambiantes o poco definidos y la necesidad de resolver situaciones con entregas que se alargan demasiado. Se suma a ello el deseo de reforzar la idea de priorizar los requerimientos desde un enfoque del negocio, orientado al cliente. Ese concepto, que tiende a diluirse en proyectos de esta naturaleza, no se pierde nunca de vista con esta metodología.

El acompañamiento de Pragma incluyó una capacitación que se estructuró en una jornada de formación teórico-práctica, en la que se presentó la metodología al grupo de desarrollo de manera didáctica y contundente. El posterior proceso de Mentoring en las primeras iteraciones resultó muy positivo y necesario, ya que nos permitió consultar y evaluar en forma conjunta cada una de las situaciones que se nos presentaron.

A pesar de que la implementación recién va por su tercera iteración, ya notamos cambios muy positivos. Nos ayudó a organizarnos de manera más eficiente, los roles están más claros, y el alcance de las tareas a cumplir, correctamente definido. El usuario sabe qué resultados esperar al finalizar la iteración y, por otro lado, advertimos que internaliza progresivamente la nueva modalidad de trabajo”.

hace cinco siglos. Luego fueron aplicados por Toyota en su principal planta de producción, hace 56 años.

Para dar una idea de la amplitud de la comunidad agile en el mundo, basta sólo con buscar en Internet la cantidad de menciones del término. Para tomar un ejemplo cuantitativo, la Scrum Alliance ya registra oficialmente más de 64.000 Certified Scrum Masters (personas que tomaron el curso oficial) y más de 2100 Certified Scrum Practicioners (personas que son CSM y que demostraron haber aplicado Scrum durante más de un año). Del lado de Extreme Programming, si bien no se consiguen números precisos de adopción, la comunidad también es muy numerosa y se habla de decenas de miles de proyectos en el mundo usando las prácticas de XP, en dominios tan variados como el de telcos, media, banca, finanzas, seguros, ERPs, defensa, Internet, salud o entretenimiento.

Basándonos en este mito y su refutación, aconsejamos:

1. Aprovechar que las prácticas ágiles están de moda para convencer a su organización de probarlas.
2. Apoyarse en las experiencias existentes para no tener que “reinventar la rueda”.
3. Hacerse acompañar por personas experimentadas y así evitar errores típicos.

Segundo acto

El otro mito imperante es: “Esto no nos va a servir, somos demasiado especiales”. Casi todas las organizaciones se consideran únicas a la hora de decidir si introducen prácticas ágiles. Creen que la estructura, el contexto, el proyecto, el equipo o los usuarios tienen características tan específicas que van a impedir la adopción. Otro argumento que se suele escuchar es que las prácticas ágiles podrán ser muy buenas para equipos chicos en un mismo sitio, pero seguramente no escalan para equipos de centenares o miles de personas distribuidas.

La realidad es que, en el mundo de los negocios actual, cualquier organización y proyecto es único y tiene su propio contexto. También es cierto que la adopción de prácticas ágiles suele chocar contra algunas características existentes de cada organización. En ámbitos con estructuras jerárquicas muy establecidas, el principio de equipos autogestionados de Scrum suele incomodar. Las prácticas ágiles (en particular Scrum, Kanban y XP) son difíciles de enmarcar en contratos con alcances fijos y detallados. Dicho esto, hay que remarcar que las prácticas ágiles requieren un cambio de paradigma y que las organizaciones más “conservadoras” en su cultura tendrán mayores dificultades a la hora de implementar dichos cambios. Es lo que remarca Stephen R. Covey en su muy pragmático principio: “Si seguimos haciendo lo que estamos haciendo, seguiremos consiguiendo lo que estamos consiguiendo”.

Respecto del escalamiento en organizaciones grandes, si bien es cierto que varias prácticas ágiles fueron pensadas para equipos de menos de diez personas, hace tiempo que existen mecanismos probados y eficientes para poder aplicarlas con éxito en grupos de cierto tamaño. Yahoo, 3M, BBC, Nokia, Primavera, Amazon, Google, Microsoft y Boeing son algunas de las empresas que aplican estos conceptos a mayor escala. Yahoo, en particular, afirma haber aplicado Scrum en un proyecto con 700 desarrolladores.

En resumen, considerando este segundo mito, recomendamos:

1. Elegir qué prácticas ágiles se adoptarán de inmediato y cuáles tendrán que adoptarse más adelante (por ejemplo, empezar por prácticas de test automatizado en integración continua de XP,

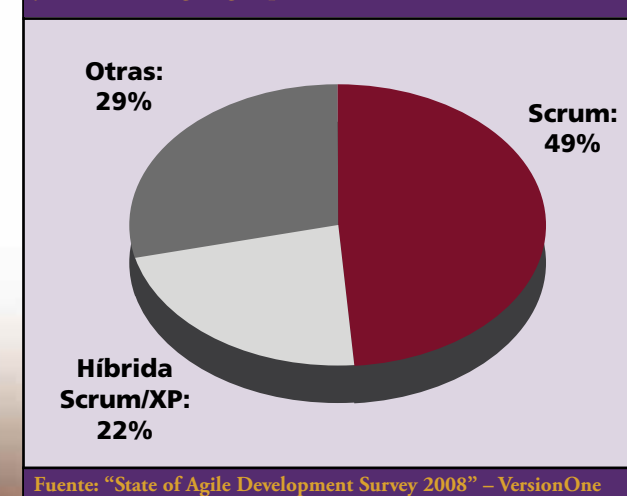
- dejando para una segunda etapa las más resistidas, como programación de a pares).
2. Implementar cada práctica a paso firme, mostrando y difundiendo los logros. Es fundamental reevaluar constantemente por qué sí o por qué no estamos implementando cada práctica.
3. Empezar con experiencias piloto en grupos reducidos para afirmar los valores y beneficios de las prácticas ágiles y luego definir formas de escalamiento hacia áreas o grupos más grandes.

Tercer acto

La última de las máximas que se escucha habitualmente es: “Agile va a solucionar tus problemas de inmediato”. Se suelen tener elevadas expectativas al implementar prácticas ágiles, esperando que sea una solución mágica que borre rápidamente problemas recurrentes en la empresa. ¿Qué nos muestra la experiencia? Que es cierto que uno puede apoyarse sobre ellas para dejar al descubierto los principales problemas de la organización para desarrollar software. La mala noticia es que el camino es difícil y que ninguna metodología agile es mágica.

Al proponer un marco con iteraciones de duración igual o menor a un mes, en el cual cada equipo compromete un incremento de funcionalidad en un software que sea de valor para el negocio, se ponen en evidencia rápidamente los impedimentos que ponen en riesgo dichos acuerdos. Scrum propone una actividad sistemática de retrospectiva al finalizar cada iteración, en la que el equipo intenta identificar los principales obstáculos que dificultaron su trabajo. Estos impedimentos pueden revelar, por ejemplo, limitaciones del equipo, problemas en la relación

¿Qué metodología ágil aplica más fielmente?



Calidad en menos tiempo

POR ILSE DIERICKX MSC, BÉLGICA*

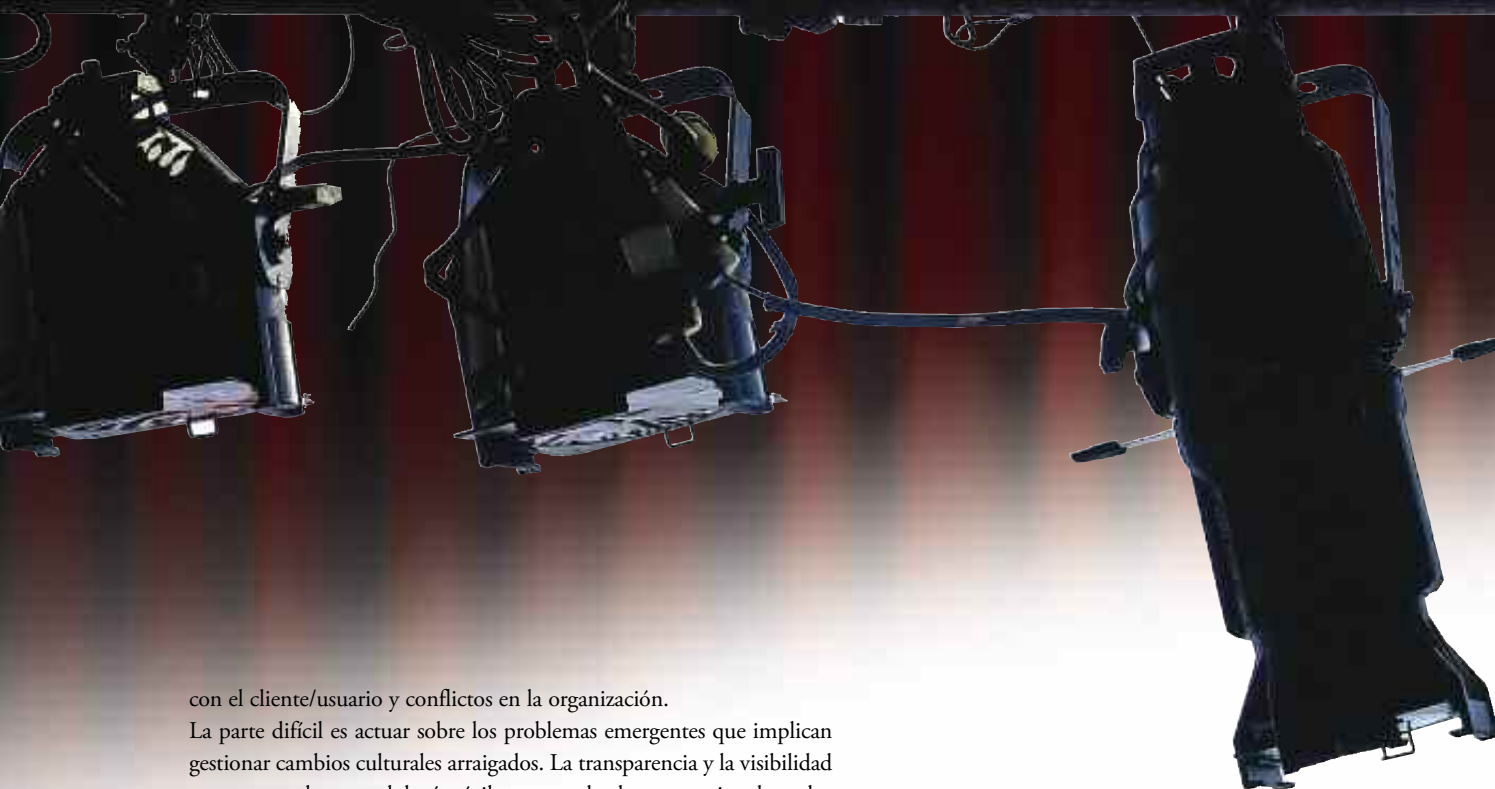
La autora trabajó –durante los últimos cuatro años– en Ventouris, un importante proyecto de desarrollo agile en el sector de la seguridad social belga. El desarrollo involucró 60 personas, atendió a ocho clientes, demandó tres años y 100 años-personas de esfuerzo.

“Ventouris empezó como un proyecto RUP tradicional, pero, una vez iniciado el desarrollo, decidimos introducir prácticas de Scrum y Extreme Programming. Lo hicimos por varias razones. El producto Ventouris es usado por ocho clientes, lo que requiere una colaboración cercana para construirlo adecuadamente. Es por eso que el *feedback* temprano es fundamental para la satisfacción del usuario. El éxito de un proyecto agile es fuertemente influenciado por la voluntad del cliente de colaborar de cerca con el equipo de desarrollo.

Como la mayoría del tiempo el cliente no sabe cómo quiere que ciertas funcionalidades se implementen, hacemos demos para generar discusiones reales sobre el producto. Para capturar todos los requerimientos del cliente usamos un Product Backlog (a la Scrum), el cual se prioriza y se divide en un Release Backlog (plan de las distintas entregas) y un Sprint Backlog (tareas de la iteración). El análisis detallado de un requerimiento del backlog y su especificación en forma de Historias de Usuario (una técnica agile para especificar requerimientos de software, enfocada en relatar el objetivo de un tipo de usuario para una funcionalidad) debe ocurrir en el momento justo, no antes. Es importante contar con usuarios reales que validen las Historias de Usuarios y que teste en la aplicación.

Ventouris también considera requisitos legales, que frecuentemente se deben implementar lo antes posible. Usar metodologías ágiles nos ayuda a manejar el cambio, y las prácticas de XP y Scrum nos permiten entregar un producto de calidad dentro del cronograma y el presupuesto estipulado”.

*Desarrolladora senior Java EE en Cegeka, un proveedor belga de soluciones IT end-to-end para la industria automotriz, bancas y seguros, y para el gobierno.



testimonios

TESTIMONIO CLIENTE

Complejidad bajo control

POR LEANDRO CANIGLIA PHD, DIRECTOR DE DESARROLLO, CAESAR SYSTEMS, BUENOS AIRES

El columnista cuenta cómo utilizan prácticas ágiles, pero sin adherir a todos sus puntos, sino adaptándolas a sus necesidades. Un ejemplo más de la flexibilidad de la metodología.

Nuestra compañía desarrolla software para la industria del petróleo desde 1996. En 2004 empezamos a aplicar sistemáticamente metodologías ágiles como XP en nuestro producto core en proyectos a medida. Por un tiempo, también incursionamos en Scrum, aunque después decidimos que no nos servía. Adecuamos varias prácticas ágiles a nuestra cultura. Partimos el trabajo de desarrollo en historias de usuarios y tareas. Hacemos bastante *pair programming*, escribimos tests unitarios de todo y los corremos en cada integración. Integramos todo todos los días, no tenemos *code ownership*, refactorizamos permanentemente, aún bajo presión de tiempo, y no hacemos horas extras. Durante una época, intentamos usar puntos para estimar las historias de cada iteración, pero después advertimos que eso creaba *overhead* administrativo. Desarrollamos una aplicación web que llamamos *Story Evaluation Tool*, en la que los *stakeholders* completan un cuestionario para cada historia, se evalúa el cuestionario numéricamente y se arroja un ranking. Las historias con puntaje más alto entran en la iteración, las otras se postergan.

Tenemos el rol del usuario, pero no es tan antojadizo como lo plantea XP, ya que el equipo de desarrollo tiene una participación activa en la discusión de historias. Creemos que es un error considerar que el usuario (o su representante) puede discernir prioridades. Lo que puede hacer es identificar una necesidad, lo cual es sólo un punto de partida. Nuestra metodología también se aparta de XP en aspectos donde nos parece insuficiente. Por ejemplo, si bien es cierto que un buen *coverage* de tests unitarios es crítico, la experiencia nos mostró que una dosis importante de QA tradicional es insoslayable. Tan es así, que trabajamos con un equipo de QA de Pragma del mismo tamaño de nuestro equipo de desarrollo.

Otro punto flojo de XP es la documentación. Coincidimos en que no es bueno ni necesario documen-

tar cómo está implementado el sistema (comentar código es definitivamente una mala práctica en la que nunca caemos), pero tenemos un equipo de tres escritores técnicos que documentan las abstracciones que hacemos, es decir, los modelos que definimos para representar el dominio. Si bien esos documentos no hacen referencia al software, permiten comprobar si un modelo de objetos es o no adecuado. Adoptar metodologías ágiles ha tenido para nosotros grandes beneficios. En primer lugar, al tener una estructura plana, todos escribimos código, tests, discutimos el diseño y hacemos *peer review*, entre otras cosas. Así, ganamos agilidad, porque todos conocemos el sistema y no hay principio de autoridad que eche a perder ninguna contribución. Lo segundo es la cultura de testing. Los tests unitarios son el capital más importante de un equipo de desarrollo. Sin un buen *coverage*, se programa con miedo de romper algo por rebote, por lo tanto, no se refactoriza y eso hace que la complejidad aumente. Lo tercero que logramos es claridad en la planificación, la delimitación de historias y tareas permite definir *workflows* y procesos que hacen nuestro trabajo más predecible, ya que se incluyen muchos puntos de control.

Pero hay desafíos que enfrentar. El más importante es cómo mantener la complejidad bajo control. Un equipo de desarrollo ágil y motivado tiene una capacidad de producción altísima, tanto que corre el riesgo de generar complejidad por acumulación. El desafío es aumentar la funcionalidad y versatilidad del sistema sin vulnerar el principio de simplicidad de Dan Ingalls: "Un sistema simple es aquél que puede ser comprendido por un solo individuo".

En 2004, Pragma Consultores organizó el Primer Workshop Argentino de Metodologías Ágiles, con más de 200 participantes.

con el cliente/usuario y conflictos en la organización.

La parte difícil es actuar sobre los problemas emergentes que implican gestionar cambios culturales arraigados. La transparencia y la visibilidad que aportan las metodologías ágiles para todos los actores involucrados (equipo, management, clientes, usuarios) resultan difíciles de "digerir" por algunas personas. Varias de las prácticas de Lean ("Eliminar los residuos", "Ver el conjunto", "Crear la integridad") permiten llevar a cabo una reflexión completa sobre el fundamento de cada actividad de un proceso, tratando de eliminar todos los pasos que no aportan valor y que suelen relacionarse con la burocracia. Estos impedimentos suelen trascender el mero ámbito del desarrollo de software e involucrar una visión transversal a todo el negocio de una empresa, implicando mejoras a articular entre múltiples sectores.

Por esto, sobre esta tercera máxima, proponemos:

1. Estar preparados para el momento en que se "saquen los trapitos al sol".
2. Trabajar en las formas de reaccionar, reflexionar y emprender iniciativas de mejora.
3. Pensar a los equipos ágiles como laboratorios donde probar variantes con un riesgo limitado.
4. Tener iteraciones cortas que permitan cometer errores con impacto moderado (una sola iteración) para aprender.

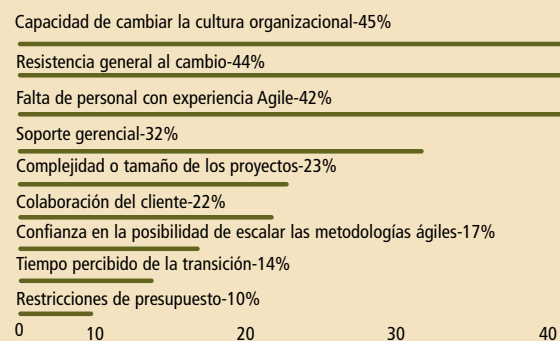
5. Ser permeables al error como mecanismo eficaz de aprendizaje. Como dice el proverbio, "Aquél cuyo pie resbala, muestra el camino para muchos".

Epílogo

El camino de adopción de las prácticas ágiles es diferente en cada organización: para algunos, es un camino complicado y largo, y para otros la transformación resulta más natural. A veces la simplicidad de las prácticas ágiles facilita una adopción rápida, y otras veces las transformaciones culturales requeridas generan resistencias difíciles de sortear. A lo largo de esta obra de tres actos, vimos varios consejos que pueden ayudarnos a obtener resultados reales y a no esperar milagros ni beneficios mágicos. De cualquier forma, la experiencia demuestra que los logros y beneficios que se construyen a través de las prácticas ágiles justifican cada paso en el camino. Así que el último consejo (¡ya fueron demasiados!) es: ¡animarse a probar!

Ahora sí, a caminar hacia la agilidad, de a un paso a la vez.

¿Cuáles son las barreras para seguir adoptando "Agile" en su organización?



Fuente: "State of Agile Development Survey 2008" – VersionOne

¿Cuál fue la razón más importante que llevó a su equipo u organización a adoptar "Agile"?

Acelerar Time-to-Market	22%
Mejorar la habilidad de manejar prioridades cambiantes	21%
Incrementar la productividad	12%
Mejorar la calidad del software	10%
Mejorar el alineamiento entre TI y negocio	9%
Mejorar visibilidad sobre proyectos	6%
Reducir riesgos	6%
Simplificar el proceso de desarrollo	4%

Fuente: "State of Agile Development Survey 2008" – VersionOne