



METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SISTEMAS

Licdo. Joseph Rodríguez Marín

MAP., Autor

RESUMEN

Cada día se escucha con más fuerza a nivel global el tema de la emergente utilización de metodologías ágiles para el desarrollo de software: versiones más recientes de libros de autores reconocidos en la ingeniería de software ya tocan este tema; además de los innumerables simposios que se realizan al respecto. Sin embargo: ¿Qué son las metodologías ágiles?, ¿qué nuevas implicaciones traen al desarrollo de software y en el cambio cultural de las organizaciones?, ¿qué tanto se conoce y que se está haciendo en Costa Rica al respecto? En el presente artículo, se pretende dar respuesta a estas interrogantes.

I. ¿QUÉ SON LAS METODOLOGÍAS ÁGILES?

El desarrollo ágil comprende un conjunto de metodologías para la creación de software, todas fundamentadas en principios comunes que giran en torno al denominado Manifiesto para el desarrollo ágil de software [1]. Este manifiesto establece que debe priorizarse la conformación del equipo por encima del entorno, las herramientas o los métodos. En consecuencia, el entorno del proyecto debe adaptarse al equipo responsable de su ejecución; se debe evitar la elaboración de documentación innecesaria, enfocándose en lo esencial; propiciar una colaboración constante entre el equipo y el cliente; y privilegiar la capacidad de adaptación frente a una adhesión estricta al plan inicial, el cual debe ser flexible y abierto, no rígido.

Asimismo, el Manifiesto se articula en torno a una serie de principios fundamentales, entre los cuales destacan: otorgar máxima prioridad a la entrega rápida de productos funcionales al cliente; acoger los cambios en los requisitos en cualquier fase del desarrollo, entendidos como una ventaja competitiva; entregar versiones operativas del software en intervalos breves; fomentar una colaboración estrecha entre desarrolladores y clientes; procurar que el software funcione desde la primera entrega; mantener un compromiso permanente con la excelencia técnica y el diseño de calidad, lo cual favorece la agilidad; y practicar la simplicidad, entendida como la capacidad de maximizar la cantidad de trabajo no realizado.

Dentro de este enfoque, las metodologías ágiles presentan diversas variantes o “sabores”, cada una con una perspectiva particular sobre cómo abordar el desarrollo de software. Entre ellas se encuentran:

- XP, “la programación extrema”, formulada por Kent Beck [2], pone más énfasis en la adaptabilidad que en la previsibilidad. Es decir, los cambios en los requisitos son inevitables y deben considerarse parte natural del proceso de desarrollo. Resulta irrealista intentar formular todos los requisitos desde el inicio, siendo lo fundamental fomentar una colaboración estrecha entre el desarrollador y el cliente. XP se basa en principios como los siguientes: el uso de historias de usuario, en las que el cliente describe brevemente, mediante tarjetas físicas, las características que el sistema debe poseer; la definición de los roles del equipo; y un proceso de desarrollo orientado a iteraciones breves, con el objetivo de entregar software funcional con rapidez. Los principios fundamentales de la programación extrema son: simplicidad, comunicación, retroalimentación y “mucho coraje”.
- Scrum, finalmente formalizado por Ken Schwaber [2], describe, mediante un enfoque ágil, la gestión y el control de desarrollos complejos de software y productos, utilizando prácticas iterativas e incrementales. SCRUM promueve el desarrollo en iteraciones breves (sprints) de no más de un mes, durante las cuales se crea un incremento de software operativo. Las características priorizadas para desarrollar (backlog) durante el sprint son definidas por el dueño del producto (Product Owner), quien las comunica al equipo. Al finalizar el sprint, el equipo presenta el incremento funcional del software.

Scrum propone la organización de equipos auto-dirigidos que realizan reuniones breves, pero altamente efectivas conocidas como scrums diarios, en las que se revisa el avance del proyecto. Durante el sprint, el cliente se integra al equipo; se elaboran planes para la identificación y mitigación de riesgos a lo largo del proceso, y se fomenta la transparencia respecto a las responsabilidades de cada miembro (“todos saben qué tiene a cargo cada uno”). Las reuniones de avance son frecuentes y deben existir mecanismos tempranos de advertencia: los problemas “no se barren bajo la alfombra”. Al concluir el sprint, se realiza una sesión de retroalimentación conocida como retrospectiva.



Fig. 1. Flujo de SCRUM [2].

- Otros métodos ágiles incluyen el Desarrollo de Sistemas Dinámico (Dynamic Systems Development Method, DSDM), Crystal Clear, Agile Unified Process (AUP), así como prácticas ágiles específicas como Test-Driven Development (TDD) y Pair Programming.

II. ¿CUÁLES NUEVAS IMPLICACIONES TRAE LA UTILIZACIÓN LAS METODOLOGÍAS ÁGILES AL DESARROLLO DE SOFTWARE Y A LAS ORGANIZACIONES?

El desarrollo ágil parte del principio de generar únicamente la documentación mínima necesaria. Las organizaciones actuales tienen arraigada la costumbre de producir documentación para todo, muchas veces de forma excesiva. El formato y volumen de esta documentación han sido, en muchos casos, definidos por requerimientos de auditoría. No sorprende, entonces, que el desarrollo ágil proponga, en primera instancia, que el entorno debe adaptarse al equipo, especialmente cuando el cliente forma parte activa de dicho equipo.

- Las metodologías: El paradigma tradicional, usualmente conocido como modelo en cascada -aún el más utilizado - da paso a iteraciones mucho más breves, en las que el cliente recibe versiones funcionales del producto en lapsos máximos de un mes, según lo establece Scrum. Esta dinámica favorece una mayor calidad del

producto, al menos desde la perspectiva de aceptación, ya que el cliente puede visualizar el producto de forma anticipada y, en consecuencia, aprobarlo o solicitar adaptaciones sin que esto implique un impacto severo en el software ya construido [3]. Las metodologías ágiles, además, resultan más comprensibles que otras más voluminosas y complejas en sus principios y procedimientos.

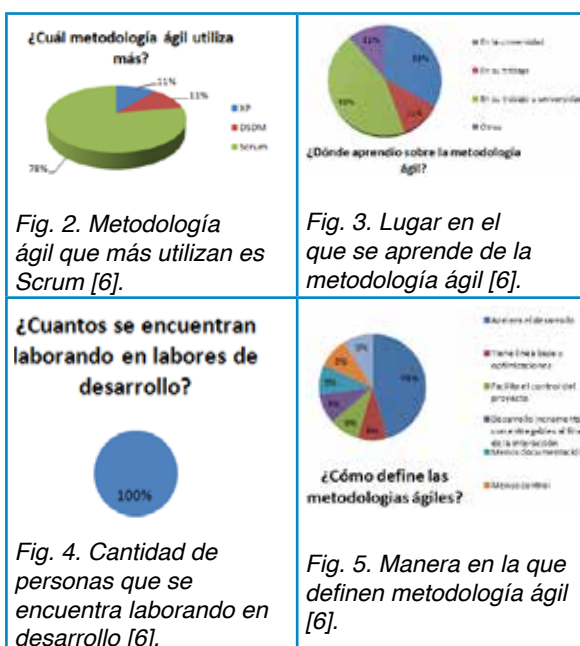
- Los estimados: Es difícil generar estimaciones realistas del proyecto si no se dispone de una visión completa de los requisitos desde el inicio -“es difícil ver el bosque cuando solo se ve un árbol”-. Este es un aspecto que debe mitigarse si se pretende utilizar metodologías ágiles, las cuales proponen iniciar el desarrollo sin conocer en detalle todos los requerimientos del sistema.
- El control de cambios: En el enfoque ágil, el cambio es considerado parte natural del proceso y, por tanto, inevitable. Esto convierte la gestión del cambio en un reto constante para los desarrolladores.
- El cliente como parte del desarrollo: En metodologías como Scrum, la participación continua del cliente en múltiples iteraciones (sprints) puede volver el proceso costoso para este, además de requerir una mayor presencia y compromiso por parte del usuario.
- Diseño del software: Los métodos ágiles promueven el uso exclusivo de la documentación esencial, lo cual, llevado al extremo, podría derivar en un diseño insuficiente. “Un problema no técnico de la metodología es cuando los clientes del sistema utilizan una organización externa para el desarrollo” [4]. Esta situación se complica si no existe documentación suficiente, lo cual dificulta, por ejemplo, la redacción de contratos entre el cliente y el proveedor en ausencia de un documento de requerimientos formal.
- Cambio cultural: Este es un aspecto que la organización debe abordar previamente si desea implementar con éxito una metodología ágil.
- Requiere valentía: Se necesita determinación para sustituir código ya desarrollado en función de cambios en los requisitos surgidos durante el proceso.

III. ¿QUÉ TANTO SE CONOCE Y QUE SE ESTÁ HACIENDO EN COSTA RICA AL RESPECTO?

De acuerdo con el Manifiesto para el desarrollo ágil de software:

En entrevista al señor Tomás Araya, encargado del Departamento de TI – Innovaplant de Costa Rica, él comenta: Estamos muy convencidos con la programación extrema ya que los tiempos de respuesta a una solicitud en cuanto un proyecto son más cortos y esto nos ahorra tener que contratar más ingenieros, y lograr desarrollar más proyectos en corto tiempo, actualmente nosotros utilizamos este método para el 100% de los proyectos y no hemos tenido problema alguno, hemos visto que los riesgos o errores son más fáciles de corregir, si presentamos primero un ejemplo al usuario final y que este decida si así está bien o desea una modificación, el método nos permite programar rápido, dejando un poco de lado la documentación [5].

Para los fines de este artículo, se llevó a cabo una breve investigación sobre el uso de metodologías ágiles entre estudiantes de la Universidad Latina, campus Heredia, en el nivel de licenciatura de la carrera de Ingeniería en Sistemas, así como en un par de otras universidades [6]. Los resultados fueron los siguientes:



En efecto, en la actualidad, las empresas e instituciones utilizan con mayor frecuencia Scrum, al tratarse de una metodología ágil para el desarrollo tanto del proyecto como del producto.

IV. CONCLUSIONES

- Cambio cultural de las organizaciones: Este es un factor fundamental para implementar con éxito las metodologías ágiles. La organización debe ser capaz de seleccionar, entre las distintas opciones disponibles, aquella metodología que mejor se ajuste a las características de su proyecto. Debe alcanzarse el principio de que el entorno se define en función del equipo, y no a la inversa.
- Documentación: El uso de metodologías ágiles no implica necesariamente la ausencia total de documentación, sino la producción de la documentación estrictamente necesaria. Lo que se debe procurar es un enfoque práctico. Desde la perspectiva de este autor, en aras de garantizar la calidad, es indispensable documentar adecuadamente los modelos requeridos.
- ¿Cero calidad?: Las metodologías ágiles, al promover principios como el trabajo en equipo, la comunicación, la simplicidad, la planificación, la mitigación de riesgos, la entrega temprana al cliente en iteraciones cortas y la clara definición de roles y responsabilidades, contienen elementos que apoyan el logro de la calidad. “XP es un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y las lleva a los extremos. Se basa en valores, principios y prácticas esenciales” [7].
- Conocimiento experto: Dado que la documentación es mínima y que se trabaja con una entrega ágil de productos en función de un orden priorizado de requerimientos, se vuelve fundamental el conocimiento previo del equipo, por ejemplo, en cuanto a reglas de negocio. Una estrategia para mitigar riesgos consiste en integrar expertos directamente en el proceso. No está de más reiterar que un factor crítico para lograr agilidad es el conocimiento especializado del propio equipo de desarrollo.
- Como en las demás metodologías, siempre se seguirá Recurso humano: Al igual que en otras metodologías, el éxito sigue dependiendo de las actitudes y aptitudes del recurso humano.

La dependencia del usuario, más crítica en el enfoque ágil, debe medirse y equilibrarse antes de iniciar el desarrollo; por ejemplo, designando usuarios expertos con capacidad de decisión. “La programación extrema es un método ágil conocido que integra una variedad de buenas prácticas de programación, como las pruebas sistemáticas, la continua mejora del software y la participación del cliente en el equipo de desarrollo” [4].

- Entrega y desarrollo rápido: En muchos casos, la entrega oportuna de sistemas nuevos puede ser más valiosa para las empresas que contar con una funcionalidad completamente detallada desde el inicio.
- Enfoque iterativo y producción: Por sus principios, las metodologías ágiles permiten al equipo enfocarse en la producción de software funcional. Sommerville [4] recomienda que, “al crecer la presión por una entrega rápida del software, se utiliza cada vez más un enfoque iterativo para el desarrollo del software como una técnica de desarrollo estándar para sistemas pequeños y de tamaño medio, especialmente en el dominio de los negocios” [4].

REFERENCIAS

[1] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland y D. Thomas, Manifiesto por el Desarrollo Ágil de Software, 2001. [En línea]. Disponible en: <https://agilemanifesto.org/iso/es/manifesto.html>

[2] Laboratorio Nacional de Calidad del Software, Metodologías de desarrollo de software. En Ingeniería del software: Metodologías y ciclos de vida, pp. 39–43, INTECO, España, 2009.

[3] S. R. Pressman, Ingeniería de software: un enfoque práctico, McGraw-Hill, 2006.

[4] I. Sommerville, Ingeniería de software, 7.ª ed., Addison Wesley, 2005.

[5] A. Céspedes, entrevista, Innovaplant de Costa Rica, 16 de julio de 2013.

[6] J. Rodríguez, Resultados encuesta: Utilización de metodologías ágiles, 2013, inédito.

[7] K. E. Kendall y J. E. Kendall, Análisis y diseño de sistemas, 6.ª ed., Pearson Educación, 2005.