

Utilidad de los procesos ágiles en el desarrollo de software

J. Dolado, Universidad del País Vasco – Euskal Herriko Unibertsitatea

D. Rodríguez, Universidad del Alcalá.

Uno de los elementos clave dentro de la ingeniería del software es la definición del proceso de desarrollo por el grupo de trabajo encargado de realizar el sistema software. Mucho se ha hablado en la literatura sobre las ventajas o desventajas de utilizar procesos clásicos, estructurados o rígidos en el desarrollo de software, frente a procesos “más desordenados” (vulgarmente hablando). Vamos a referirnos a continuación a dos aspectos importantes sobre la utilidad de los procesos ágiles: los aspectos críticos de los mismos y la interacción con la arquitectura software. Parecería que los proponentes de una arquitectura software rígida serían contrarios a una propuesta de métodos ágiles para el desarrollo, pero ambos puntos de vista no están enfrentados y van apareciendo las primeras evidencias sobre la utilidad de las metodologías de desarrollo ágiles. Mientras que desde el punto de vista teórico el uso de procesos ágiles estaría enfrentado con determinados productos software “estructurados”, recientes estudios nos muestran que no hay contradicción entre ambas posturas y que los distintos aspectos pueden ser reconciliados en la práctica.

Como el mismo término “ágil” indica, los procesos software que se engloban bajo ese término poseen las características de flexibilidad y de respuesta a condiciones cambiantes. Los supuestos básicos de los métodos ágiles son: prestar más importancia a los individuos y sus interacciones que a los procesos y las herramientas, preocuparse más sobre el funcionamiento del software que sobre la documentación, tener una mayor colaboración con el cliente que sobre las cuestiones formales y, sobre todo, orientarse a una respuesta inmediata frente al seguimiento de un plan estricto.

Los métodos ágiles constituyen una reacción a los métodos tradicionales planificados que enfatizan un enfoque excesivamente ingenieril y racional y que asumen que todos los problemas se pueden especificar completamente. El enfoque “tradicional” propugna una planificación intensa, procesos estrictos y rigurosa reutilización, de tal modo que la actividad de desarrollo se asemeja a una ingeniería estricta. Por el contrario el enfoque “ágil” promueve una respuesta rápida a entornos cambiantes, cambios en los requisitos y modificaciones en los plazos de entrega. Los acrónimos que se incluyen bajo el paraguas de método ágil son variados e incluyen a términos como Extreme Programming -XP-, Scrum, Crystal, Lean Software Development -LD- y otros.

Las principales críticas que se atribuyen a los enfoques ágiles son las siguientes: a) sus ideas no

son nada innovadoras, pues es lo que se hacía desde hace tiempo b) la falta de enfoque hacia la arquitectura del software va a generar productos de menor calidad y c) no existen demasiadas pruebas de la utilidad de estos métodos, además de la falta de replicabilidad de esas prácticas.

Un reciente estudio bastante sistemático realizado por Dyba y Dingsoyr ha mostrado algunos de los beneficios y limitaciones de la introducción de métodos ágiles en el desarrollo. Así, se ha encontrado que XP ha sido más difícil de introducir en organizaciones complejas y que era más adecuado para pequeños grupos que para grandes proyectos. La adopción de los métodos ágiles es fácil en muchos casos y se encuentran beneficios en la colaboración con el cliente, en el tratamiento de los errores y en algunos aspectos de gestión, incluso en la estimación. También se encuentra cierta mejora en la percepción de los clientes sobre los efectos de la comunicación, aunque si el contacto es muy continuo se percibe cansancio. Igualmente se puede destacar que el uso de metodologías ágiles va acompañado de distintas prácticas de gestión de proyectos. Por ejemplo, dado que los métodos ágiles son dependientes de las personas, un modelo de gestión que permita intercambiar personas presentará grandes limitaciones. La productividad medida en líneas de código también aumenta si se utiliza XP. Con respecto a la calidad del producto, ese estudio indica, sin grandes convicciones, que la calidad del producto puede incrementarse, pero no es concluyente. Tampoco se puede establecer una correlación entre la satisfacción en el trabajo de los programadores y las prácticas ágiles.

Los factores de éxito con evidencias cuantitativas en la implementación de este tipo de procesos son: la utilización de una estrategia correcta de entrega de los productos, el uso de una práctica adecuada de las técnicas de ingeniería del software ágiles, la integración de un grupo de trabajo de alta cualificación junto con una alta implicación del cliente y un proceso de gestión adecuado. El principal problema para la gestión de los métodos ágiles es la correcta cuantificación y la evaluación del estado real del proyecto. No existen muchas conclusiones absolutamente claras y generalizables, pero podemos decir que no se ha detectado ninguna evidencia en contra de la utilización de estas metodologías ágiles. Quizá el punto más delicado sería el hacer compatible los conceptos de “arquitectura software” con el de “agilidad”. La arquitectura del software, definida según el Rational Unified Process, se define como “el conjunto de decisiones sobre la organización de un sistema software, la selección de los elementos estructurales y los interfaces de los que se compondrá el sistema, junto con el comportamiento que especifican esos elementos. La arquitectura del software no se preocupa sólo de la estructura y del comportamiento, sino también del uso, de la funcionalidad, del rendimiento, reutilización, comprensibilidad y de las restricciones económicas y tecnológicas”.

Diversos autores indican que estos conceptos no son incompatibles puesto que se puede construir una arquitectura con distintos niveles de detalle, de tal modo que quienes utilicen métodos

ágiles puedan establecer sus requisitos de modo flexible y con una comunicación adecuada. Además, ya existen propuestas “ágiles” que no reniegan de la existencia de una documentación sobre la arquitectura. El mayor problema reside en los requisitos no funcionales, que son los que quedan al margen en las propuestas ágiles. Sin embargo ya existen alternativas para expresar los conceptos más importantes de una arquitectura e incorporarlos a la comunicación de los grupos de desarrollo y ya se ha hecho para Scrum. Podemos concluir que aunque no existan resultados concluyentes sobre la superioridad de los métodos ágiles, tampoco se evidencian desventajas con respecto a los métodos clásicos en los aspectos críticos del desarrollo de arquitecturas software.

Referencias

- Abrahamsson, P.; Babar, M.A.; Kruchten, P.; , "Agility and Architecture: Can They Coexist?," *Software*, IEEE, vol.27, no.2, pp.16-22, March-April 2010
- Chow, Tsun y Dac-Buu Cao, "A survey study of critical success factors in agile software projects", *The Journal of Systems and Software* 81 (2008) 961–971
- Dyba, Tore y Torgeir Dingsøyr, "Empirical studies of agile software development: A systematic review", *Information and Software Technology* 50 (2008) 833–859
- Falessi, D., Cantone, G., Sarcia', S.A.; Calavaro, G., Subiaco, P.; D'Amore, C.; "Peaceful Coexistence: Agile Developer Perspectives on Software Architecture," *Software, IEEE*, vol.27, no.2, pp.23-25, March-April 2010