

Adopción de prácticas ágiles de desarrollo de software en los planes de estudio de universidades de Costa Rica: revisión de la literatura

Carlos Martín Flores González, Escuela de Ingeniería en Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
Email: martin.flores@ieee.org

Resumen—Aquí va el resumen

Palabras Clave—IEEEtran, journal, L^AT_EX, magnetic, paper, template.

I. INTRODUCCIÓN

EN las últimas décadas se ha hecho un esfuerzo significativo en identificar buenas prácticas, modelos y métodos que conduzcan a desarrollar software de forma más eficiente. Los desarrolladores de software tienden a clasificar las metodologías de desarrollo en dos categorías [5]:

1. Metodologías de desarrollo clásicas: requieren definición de requerimientos por adelantado, documentación y planes detallados. Dos ejemplos relevantes de son el modelo de cascada y espiral.
2. Metodologías ágiles: a menudo se llama “livianas”. Esta categoría incluye *eXtreme Programming* (XP) y *Scrum*.

El movimiento de desarrollo ágil de software nace como una alternativa a los métodos tradicionales con los que se hace software, en los cuales los ciclos de desarrollo y entrega tienden a ser muy prolongados. El desarrollo ágil de software se define en el Manifiesto Ágil [4] como un conjunto de doce principios.

II. DESARROLLO ÁGIL DE SOFTWARE

Las metodologías de desarrollo ágil de software emergen al final de los de la década de los noventa. El término “ágil” se utiliza para agrupar una serie de métodos tales como *Scrum*, *eXtreme Programming*, *Crystal*, *Feature Driven Development* (FDD), *Dynamic Software Development Method* (DSDM) y *Adaptive Software Development* [6]. Las metodologías ágiles se caracterizan por promover el desarrollo iterativo e incremental y la entrega frecuente de funcionalidad prioritaria para los clientes. Estas metodologías están dirigidas a equipos pequeños y altamente colaborativos. *Scrum* y XP son las metodologías que presentan mayores niveles de adopción [8]: XP se centra en prácticas de desarrollo mientras que *Scrum* cubre principalmente la administración del proyecto.

Este documento fue realizado durante el curso de Ingeniería de Software, impartido por el profesor Rodrigo Bogarín. Programa de Maestría en Computación, Instituto Tecnológico de Costa Rica. Segundo Semestre, 2017.
Recibido el 21 de setiembre del 2017.

Las metodologías ágiles son adecuadas para proyectos con requerimientos altamente cambiantes, se motiva aceptar y responder ante los cambios. En 2001, los desarrolladores de varios de estas metodologías ágiles escribieron el Manifiesto Ágil [4] (Apéndice A). Los principios detrás del Manifiesto Ágil incluyen entrega rápida, frecuente consistente y continua de software funcional, respuesta a requerimientos cambiantes, comunicación efectiva y equipos motivados con capacidad de auto organizarse.

II-A. Prácticas Ágiles de desarrollo de software

De acuerdo con [2], a pesar de la gran cantidad de literatura disponible sobre metodologías ágiles de software, mucha de la misma mencionan poco acerca de las prácticas ágiles de desarrollo de software. La mayoría se centran principalmente en gestión, procesos y estimación, y no tanto en la parte relacionada con la ingeniería de software.

En contraste con las metodologías, las prácticas ágiles están un nivel por debajo debido a que estas son una parte muy específica de una metodología que aborda varios aspectos. Algunos ejemplos conocidos son programación en parejas y reuniones diarias. A pesar de no haber una definición común en la literatura de práctica ágil [1], se puede tomar XP en consideración para tener un mejor entendimiento por medio del estudio de la colección de prácticas de desarrollo que se promueven en esta metodología. A diferencia de otras metodologías como *Scrum*, XP se dedica tanto a la gestión del proyecto como también en la forma en cómo los equipos construyen código [7].

Para efectos de esta revisión, se define un conjunto de prácticas. Esto porque las metodologías ágiles llaman a sus prácticas de forma diferente. Para obtener un conjunto de prácticas de referencia, se tomó como punto de partida el reporte de Version One [8] el cual expone las prácticas ágiles en desarrollo de mayor aceptación. Las prácticas expuestas en este reporte se confrontaron con la literatura acerca de XP [2], [3], [7]. El resultado es la siguiente lista de prácticas ágiles en desarrollo de software:

- Pruebas
 - Pruebas unitarias
 - Desarrollo orientado a pruebas (TDD, por sus siglas en inglés)

- Desarrollo orientado al comportamiento (BDD, por sus siglas en inglés)
- Desarrollo orientado a pruebas de aceptación (ATDD, por sus siglas en inglés)
 - Integración continua
 - Estándares de codificación
 - Refactorización de código
 - Puesta en producción continua
 - Propiedad compartida del código
 - Diseño emergente
 - Control de versiones



Martín Flores es Ingeniero en Informática de la Universidad Nacional. Actualmente, realiza sus estudios de Maestría en Ciencias de la Computación del Tecnológico de Costa Rica. Sus principales intereses son: lenguajes de programación, ingeniería de software y DevOps.

II-B. DevOps

III. ENSEÑANZA DE PRÁCTICAS ÁGILES

IV. ENSEÑANZA DE PRÁCTICAS ÁGILES EN COSTA RICA

V. CONCLUSIÓN

La conclusión

APÉNDICE A EL MANIFIESTO ÁGIL

Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

REFERENCIAS

- [1] Philipp Diebold y Marc Dahlem. 2014. *Agile practices in practice: a mapping study*. En Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14). ACM, New York, NY, USA, , Artículo 30 , 10 páginas. DOI: <http://ezproxy.itcr.ac.cr:2075/10.1145/2601248.2601254>
- [2] Neil Ford. 2011 *Agile Engineering Practices*. Video. O'Reilly Media Inc. Obtenido de <https://www.safaribooksonline.com/library/view/neal-ford-on/9781449314439/>. ISBN: 9781449314439
- [3] Kent Beck, Cynthia Andres. 2004. *Extreme Programming Explained*. Addison-Wesley Professional.
- [4] Kent Beck, Mike Beelde, Arie van Bennekum, Allistair Cockburn, et al. 2001. *Manifesto for Agile Software Development*. Agile Alliance.
- [5] Li Jian y Armin Eberlein. 2009. *An analysis of the history of classical software development and agile development*. Proceedings of the 2009 International Conferencen on Systems, Man, and Cybernetics. San Antonio, TX, USA. Octubre 2009. DOI: <https://ezproxy.itcr.ac.cr:2878/10.1109/ICSMC.2009.5346888>
- [6] Rashina Hoda, Philippe Kruchten, James Noble y Stuart Marshall. 2010. *Agility in context*. SIGPLAN Not. 45, 10 (Octubre 2010), 74-88. DOI: <https://ezproxy.itcr.ac.cr:2878/10.1145/1932682.1869467>
- [7] James Shore y Shane Warden. *The Art of Agile Development*. 2008. O'Reilly Media, Inc.
- [8] Version One. *11th Annual State of Agile Report*. 2017. Obtenido de <http://stateofagile.versionone.com/>