

Integración de Arquitectura de Software en el Ciclo de Vida de las Metodologías Ágiles. Una Perspectiva Basada en Requisitos

Mg. Mirta E. Navarro¹, Mg. Marcelo P. Moreno², Lic. Juan Aranda³, Lic. Lorena Parra⁴,
Lic. Jose R. Rueda⁵, Juan Cruz Pantano⁶

Departamento de Informática - F.C.E.F. y N. - U.N.S.J.

Complejo Islas Malvinas. Cereceto y Meglioli. 5400. Rivadavia. San Juan

¹mirtaenavarro@yahoo.com.ar ²mpmoren@gmail.com ³juanaranda@live.com ⁴lorenaparra152@yahoo.com.ar
⁵josericardorueda@hotmail.com ⁶juancruz871@hotmail.com

Resumen

Las metodologías Ágiles se centran en el trabajo en equipo, la adaptabilidad y colaboración dentro del grupo de software y también entre los miembros del grupo y los usuarios finales. El uso de las Metodologías Ágiles (MA), ha marcado una tendencia [1] en su adopción al desarrollo de proyectos de software dado las necesidades cambiantes y la espera de beneficios en el menor tiempo posible. En general, pero también desde la perspectiva de los requisitos, esto hace que las MA, típicamente eviten un trabajo inicial sustancial, suponiendo que los requisitos siempre cambian y continúan cambiando a lo largo del ciclo de vida del proyecto.

La Arquitectura de Software (AR), en tanto, es una manifestación de decisiones de etapas muy tempranas del diseño sobre un sistema [2]. Estas decisiones tempranas llevan un peso importante con respecto al desarrollo del resto de un sistema, ya que condicionan otras decisiones que siguen, y en el caso de cambios, implican ramificaciones posteriores. Esto supone una captura de requisitos que no tengan cambios sustanciales en las etapas intermedias y finales del desarrollo de un proyecto.

Este tratamiento con enfoques diferentes en las primeras etapas (y también en otros aspectos), ha sido uno de los factores que ha causado la sensación de que las MA y la AS van en direcciones diferentes y no pueden coexistir juntas [3].

Sin embargo, en los últimos cinco años esta tendencia esta cambiando [4,5], hasta el punto

que ha surgido el concepto “*Arquitectura Ágil*” (AA). En la AA, se enfatiza fuertemente en el concepto de los “*Requisitos Significantes para la Arquitectura*” (RSA) o “*Architecturally Significant Requirement*” ASR, por sus siglas en inglés. En ese sentido, esta línea de investigación tiene como objetivo indagar y validar el alcance de hasta dónde pueden confluir la AS con las MA, poniendo foco en la captura de requisitos y en los procesos de identificación de los RSA.

Palabras clave: *Software Architecture, Agile methodologies, Information Systems Architecturally Significant Requirements (ASR)*

Contexto

El presente trabajo se encuadra dentro del área de I/D de la IS y de los SI, y forma parte de una etapa del proyecto de investigación: “*Integración de Metodologías Ágiles y Arquitecturas de software en el Desarrollo de Sistemas de Información*”, presentado en WICC 2016 [6] ejecutado a partir de Enero de 2016, con una duración de dos años y que tiene como unidades ejecutoras al Departamento e Instituto de Informática de la FCEfYN de la UNSJ. El proyecto se encuentra acreditado y financiado por la Secretaria de Ciencia y Técnica UNSJ.

El grupo de investigación tiene una trayectoria de 16 años en diferentes proyectos vinculados a Metodologías de Desarrollo y Tecnologías, con numerosas publicaciones en

diferentes ámbitos, y con la formación de recursos humanos en el área de interés.

Introducción

El lugar que ocupa la AS en el ciclo de vida de desarrollo de software es difícil de definir. Pero, puesto que las arquitecturas describen el espacio de solución de un sistema, tradicionalmente se la ha considerado como una parte temprana de la fase de diseño. Si bien la AS puede ser vista desde diferentes focos contextuales y de diferentes niveles de abstracción, el interés de este trabajo está centrado en los aspectos relacionados en los requisitos. Así, desde esa mirada, la AS, debe garantizar que la arquitectura sea realizable mediante una implementación adecuada que satisfaga los requisitos del sistema. En las MA, es justamente allí, en la consideración de los requisitos, donde los aspectos arquitectónicos se suelen pasar por alto. En ese sentido, normalmente, se identifican y captan los requisitos, tanto funcionales como no funcionales, y luego se intenta definir una arquitectura de software que los cumpla.

Sin embargo, esta es una tarea compleja en particular con la identificación de los requisitos no funcionales y con los requisitos mal elicitados (que no consideran aspectos AS) y puede derivar en una realización arquitectónica inadecuada. Además, se debe considerar que algunos requisitos tienen un efecto mucho más profundo en la arquitectura que otros. En este escenario, es donde cobra fuerza un concepto reciente que va de la mano con la Arquitectura Ágil, los RSA, que son requisitos que tendrán un efecto profundo en la arquitectura, y que como postula [4], si llegasen a faltar, la Arquitectura final resultante será muy diferente.

Los Requisitos de Importancia Arquitectónica

Los requisitos de software se dividen principalmente en requisitos funcionales y

requisitos no funcionales. Los requisitos funcionales corresponden a las características deseadas de un sistema; en tanto que no funcionales especifican las propiedades requeridas de un sistema. Los requisitos no funcionales pueden estar caracterizados por diferentes factores tales como pueden ser los atributos de calidad, de seguridad, de disponibilidad, restricciones del sistema, metas, características de usabilidad, etc.

Un Requisito de Importancia Arquitectónica es un requisito que tendrá un importante efecto en la arquitectura, y que como se dijo, si están ausentes, la arquitectura resultante será totalmente diferente. Tomando la definición de [7] los RSA *“son aquellos requisitos que tienen un impacto medible en una arquitectura de sistemas de software”*. Por lo tanto, no sería posible diseñar una arquitectura adecuada, si no se elicitaban adecuadamente los RSA. En general, en la identificación y especificación de los RSA intervienen las principales partes interesadas del sistema, como los usuarios finales, los desarrolladores, los administradores y los mantenedores.

El problema es que los RSA generalmente toman la forma de requisitos no funcionales, pero en algunos casos también toman la forma de requisitos funcionales. Además, los RSA suelen ser subjetivos, relativos y también interactúan entre ellos. Son subjetivos, porque pueden ser vistos, interpretados y analizados de manera diferente por diferentes personas y en diferentes contextos; son relativos, porque la importancia de cada RSA se determina a veces a partir de su relación con otros RSA en un contexto dado; y se considera que interactúan entre ellos en el sentido de que al intentar alcanzar un RSA particular, puede a su vez afectar (en forma favorable o desfavorable) a otros RSA. Los RSA son más difíciles de comprender que los requisitos funcionales [7], por lo que en general pasan desapercibidos o no obtienen suficiente atención por adelantado, factor que es más acentuado cuando se utilizan MA, donde generalmente se expresan (muchas veces en forma contradictoria) de manera

informal durante el análisis de requisitos. También son difíciles de validar cuando el proyecto está finalizado. Por ello, la identificación de los RSA, es una tarea difícil, que lleva mucho trabajo y es poco clara para los que no son expertos en arquitecturas.

Para abordar esta situación, algunos autores [4,5 y 7] han propuesto una serie de métodos y estrategias de identificación y captura de RSA a partir de documentos de requisitos tradicionales de una MA, los que serán utilizados por este grupo para analizarlos y evaluarlos.

Estrategias de captura de RSA.

Las formas de obtener RSA a partir de los documentos de requisitos, son varias. Chen [7] propone un marco para caracterizar los RSA, sobre la base de un estudio empírico que realizó con expertos. Sus hallazgos pueden mejorar el entendimiento de los requisitos y las interacciones con la arquitectura. En general, las estrategias que propone son:

- Categorizar las decisiones de diseño de arquitectura que los arquitectos de software tienen que tomar. Estas decisiones toman la forma de tablas resumidas. A partir de allí proporcionan una lista de requisitos para buscar cual de ellos podría afectar a ese tipo de decisión. Si un requisito afecta a alguna decisión de diseño, es por definición un RSA.
- Entrevistas a interesados.
- Comprensión de objetivos de negocios.

En tanto que [4 y 5] proponen utilizar enfoques tales como escenarios para caracterizar diferentes atributos, en particular los de calidad, y también plantean utilizar aproximaciones basadas en frameworks de evidencias.

Finalmente debe quedar claro que las consideraciones sobre requisitos RSA, no alcanzan a especificar la totalidad sobre las arquitecturas, y que un trabajo adicional deberá ser realizado para la adecuada especificación de las AS.

Tópico de investigación, Desarrollo e Innovación

Como se ha manifestado en [6], los posibles beneficios de una integración de Arquitectura de Software en las Metodologías Ágiles no es un tema lo suficientemente explorado. Sin embargo [4 y 5], han propuesto últimamente algunas publicaciones relacionadas en lo que se denomina Arquitectura Ágil, donde exploran con detenimiento los RSA.

Esta línea de investigación toma como punto de partida los hallazgos de [4,5 y 7] que son la base para discutir y llevar a cabo más investigaciones sobre RSA. Pensamos que esos métodos y enfoques pueden ser complementarios entre sí, y a la vez podrían ser combinados con los hallazgos que han sido ampliamente investigados por los autores de este artículo, en el área de las metodologías ágiles.

Finalmente los aportes que pudieran surgir, serán utilizados como guías para elicitar, analizar implementar y evaluar situaciones reales en proyectos de tesis y tesinas que han surgido a partir de inquietudes emergentes en proyectos de investigación anteriores que este grupo llevó a cabo [8 y 9].

Resultados y Objetivos

Los resultados que se esperan obtener al finalizar la investigación son:

- Analizar las características particulares de la relación de los requisitos tradicionales con los RSA.
- Evaluar las propuestas de [4,5 y 7], utilizando documentos de requisitos tradicionales de las MA proponiendo, eventualmente, alternativas que surjan de la línea de investigación, y aplicar y evaluar esos hallazgos en un contexto real en tesis o tesinas.
- Continuar con la línea trazada en el proyecto [10], cuyo objetivo principal es establecer la incorporación de arquitecturas

al proceso de desarrollo dirigido por metodologías ágiles, con el propósito de que esa integración favorezca el diseño de Sistemas de Información.

Formación de Recursos Humanos

El equipo de trabajo está compuesto por cinco docentes-investigadores, y seis alumnos adscriptos. En el periodo 2015-2016 se ha asesorado: una tesis de grado finalizada y cuatro tesis de grado en proceso, una de las cuales se está desarrollando en el nuevo tópico de investigación que se presenta en este trabajo.

Además, se espera realizar direcciones, a los alumnos adscriptos, en diversos trabajos orientados al desarrollo de sistemas con metodologías ICONIX-SCRUM, adoptando el enfoque de integración con arquitecturas desde la perspectiva de los RSA.

Con los resultados de la presente investigación, se harán actividades de divulgación en publicaciones y presentaciones en eventos nacionales e internacionales y también en cursos de postgrado y actualización.

Referencias

- [1] Canós J., Letelier P. “Metodologías Ágiles en el Desarrollo de Software”. JISBD 2003. España.
- [2] Urquiza Yllescas, J.F., et al, “Las Metodologías Ágiles y las Arquitecturas de Software”. Coloquio Nacional de Investigación en Ing. De Sofá. 2010, León, México.
- [3] Nord RL, Tomayko JE. “Software architecture-centric methods and agile development.” IEEE Software 2006; 23(2):47–53.
- [4] Babar MA, Brown AW, Mistrik I, Agile Software Architecture Aligning Agile Processes and Software Architectures. 2013 ISBN 978-0-12-407772-0.
- [5] Bass L, Clements P, Kazman R. Software architecture in practice 3rd ed. Addison-Wesley ISBN 978-0-321-81573-6.

- [6] Navarro, M; Moreno, M; Aranda, J; Parra, L; Rueda, J. “Integración de Metodologías Ágiles y Arquitecturas de software en el Desarrollo de Sistemas de Información.” XVIII Workshop de Investigadores en Ciencias de la Computación. 2016, Concordia, Entre Ríos.
- [7] Chen L, Babar MA, Nuseibeh B. Characterizing architecturally significant requirements. IEEE Software 2013; 30:38–45.
- [8] “Aplicabilidad de Metodologías y Tecnologías en el desarrollo de Sistemas de Información” Cod: 21/E 979. FCEF. Navarro, Mirta, et al. 2014-2015
- [9] “Convergencia de Tecnologías Informáticas y Metodologías para la Implementación de Sistemas de Información” Cod: 21/E/871 FCEF. Navarro, Mirta, et al. 2012-2013.
- [10] “Integración de Metodologías Ágiles y Arquitecturas de software en el Desarrollo de Sistemas de Información” Cod: 21/E 1027. FCEF. Navarro, Mirta, et al. 2016-2017