

# Selección de Metodologías Ágiles e Integración de Arquitecturas de Software en el Desarrollo de Sistemas de Información

Mg. Mirta E. Navarro<sup>1</sup>, Mg. Marcelo P. Moreno<sup>2</sup>, Lic. Juan Aranda<sup>3</sup>, Lic. Lorena Parra<sup>4</sup>,  
Lic. José R. Rueda<sup>5</sup>, Juan Cruz Pantano<sup>6</sup>

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

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

<sup>1</sup>mirtaenavarro@yahoo.com.ar <sup>2</sup>mpmoren@gmail.com <sup>3</sup>juanaranda@live.com <sup>4</sup>lorenaparra152@yahoo.com.ar  
<sup>5</sup>joseruardorueda@hotmail.com <sup>6</sup>juancruz871@hotmail.com

## Resumen

La Ingeniería de Software (IS) se ha convertido imprescindible en el ámbito organizacional, su desenvolvimiento y gestión depende en gran medida de los Sistemas de información (SI) y de las Tecnologías de la información (TI).

En los últimos años, se ha impuesto el uso de las Metodologías Ágiles (MA), marcando una tendencia en su adopción al desarrollo de proyectos de software. La causa principal es que en ambientes donde las necesidades de las organizaciones y la tecnología cambian rápidamente, las metodologías tradicionales predictivas han demostrado ser poco eficientes para atender los requerimientos de clientes y usuarios, limitando la competitividad y a la obtención de mayores beneficios en la producción de bienes o en la prestación de servicios, en el menor tiempo posible, y es en ese escenario donde las MA han ganado bastante

propiciados en la IS, el desarrollo de la AS es una práctica poco común para algunos desarrolladores, en especial si se utilizan metodologías de desarrollo ágiles, en las que esas actividades no se consideran relevantes.

En este trabajo se presenta un avance de la exploración, comparación y selección bajo distintos criterios de dos metodologías ágiles, con el propósito de incluir actividades de diseño de arquitecturas de software, que permita generar un modelo genérico de integración, que pueda ser aplicado a otras metodologías bajo el dominio agilista, sin dejar de lado las consideraciones de calidad, riesgo y costos asociados a la integración pretendida.

**Palabras clave:** *Software Architecture, Agile methodologies, Information Systems*

## Contexto

El presente trabajo se encuadra dentro del trabajo de grado presentado al Servicio de Difusión de la Creación Intelectual e los SI, aborda el análisis bajo distintos criterios y selección de metodologías ágiles, uno de los dos ejes fundamentales del proyecto de investigación: *“Integración de Metodologías Ágiles y Arquitecturas de software en el Desarrollo de Sistemas de Información”*, presentado en WICC 2016 [2]. El proyecto se encuentra acreditado y financiado por la Secretaria de Ciencia y Técnica Universidad Nacional de San

View metadata and citation for this article at: [www.cobice.org.ar](http://www.cobice.org.ar)  
proyectos a corto plazo, en especial, aquellos proyectos en donde los requisitos están cambiando constantemente.

La Arquitectura de Software (AS) comprende elementos de software, las propiedades externamente visibles de aquellos elementos y su interrelación para satisfacer la funcionalidad y requerimientos deseados. Si bien en los últimos años, ha comenzado a cobrar una mayor importancia dentro de los estudios

Juan (UNSJ). Iniciado en Enero de 2016, con una duración de dos años, que tiene como unidades ejecutoras al Departamento e Instituto de Informática de la Facultad de Ciencias Exactas, Físicas y Naturales de la UNSJ. El grupo de investigación tiene una trayectoria de varios 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

La Ingeniería de Software se ha convertido en una disciplina imprescindible debido a que la mayoría de los proyectos de Sistemas de o atributo de calidad, esto ha motivado la integración de actividades, técnicas y patrones de arquitectura en procesos de desarrollo ágiles.

Esta idea de integración de metodologías ágiles y arquitectura de software constituye el objetivo principal del Proyecto de investigación referenciado. Para su consecución se plantearon actividades de análisis, comparación y selección de metodologías ágiles, exploración de arquitecturas y propuesta de un modelo de integración.

## Metodologías Ágiles

Son metodologías adaptativas, que permite llevar a cabo, proyectos de desarrollo de software, adaptándolo a los cambios como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo software y, permitiendo de este modo, una mejor adaptación al entorno, maximizando la inversión y reduciendo los costos, ya sea para variar parte de su funcionalidad, añadir otra nueva, o por ejemplo, adaptar el sistema a un nuevo dominio de aplicación[1]. Las metodologías ágiles se caracterizan por el desarrollo iterativo e incremental, las entregas frecuentes, la

Información que se impulsan dependen en gran medida de la aplicabilidad de Metodologías de la elección de la AS y de las actuales características dinámicas y variables de las organizaciones que están reconsiderando las bases sobre las que se sustenta el desarrollo software.

Para el desarrollo de software existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso, desde las metodologías tradicionales predictivas hasta las actuales metodologías giles que se han impuesto en el desarrollo de proyectos de software, marcando una tendencia su adopción.

Por otra parte las arquitecturas en el desarrollo de software están cobrando cada vez mayor importancia con

simplicidad de la implementación; priorizan los requisitos, comunicación continua con el cliente y la cooperación entre los desarrolladores.

Estas metodologías constituyen una solución con una elevada simplificación, que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Por lo tanto, para ejecutar un desarrollo exitoso con metodologías ágiles, es importante comprender los requisitos del cliente, intentando extraer al máximo los deseos del usuario para entregar un producto lo más cercano a la realidad. Al ser flexibles y colaborativas, se ajustan fácilmente a cambios de requisitos, de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente y la capacidad de entrega continua y en plazos breves permite al cliente verificar in situ el desarrollo del proyecto y familiarizarse con la funcionalidad del producto progresivamente y comprobando si el sistema cumple con los requerimientos, mejorando de esta forma su satisfacción. [2].

Además, el desarrollo en ciclos de corta duración favorece que los riesgos y dificultades se repartan a lo largo del desarrollo del producto, principalmente al comienzo del mismo que permite analizar las dificultades.

Teniendo en cuentas, características, fases, colaboración con el cliente, iteraciones desarrollo incremental, elementos, ciclo de vida, roles y otros factores preponderantes en la comparación realizada de ocho metodologías ágiles surge la elección de ICONIX y SCRUM, para efectuar la integración de una arquitectura de software flexible y adaptable a dichas metodologías, que permita construir un sistema o producto de calidad. Las mencionadas metodologías se ajustan fácilmente a cambios de requerimientos, de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente y la capacidad de entrega continua y en plazos breves permite al cliente verificar in situ el desarrollo del proyecto y familiarizarse con la funcionalidad del producto progresivamente y comprobando si el sistema cumple con los requerimientos, mejorando de esta forma su satisfacción. [2]. [3].

## Arquitectura de Software.

La Arquitectura de Software reúne todos los requerimientos técnicos y operacionales, se compone de elementos de software, las relaciones entre ellos, las propiedades de ambos (elementos y relaciones) y de patrones arquitectónicos que guían a esta organización. Estas incumbencias van más allá de los algoritmos y estructuras de datos de los programas. [4] La arquitectura es principalmente importante para satisfacer los requisitos no funcionales, que están relacionados a los atributos de calidad como el rendimiento, seguridad y escalabilidad.

Los componentes fundamentales, en la AS, son los patrones arquitectónicos, siendo un conjunto de principios que proporcionan un marco para el diseño, proveen la forma a una aplicación mejoran la partición, ayudan a definir las características básicas y de comportamiento de una aplicación. Por ello, es necesario

investigar, analizar y conocer claramente las características, fortalezas y debilidades de cada patrón de arquitectura y seleccionar aquel que se adapte a la solución de un problema determinado.

Dado la importancia que existe sobre el papel de la AS en las MA, los autores Edwin Rafael Mago y Germán Harvey Alférez [11] presentan una interesante propuesta para gestionar la arquitectura de software en SCRUM, tal propuesta consiste en adicionar un sprint inicial llamado "*Sprint 0*" al inicio del ciclo de desarrollo para analizar y diseñar la generalidad del sistema, que satisfaga los requisitos y sea entendible por los miembros del equipo desde sus diferentes puntos de vista durante el desarrollo. Un punto clave, es reutilizar artefactos de software creados a partir de la arquitectura para ser más ágiles en el desarrollo de productos específicos. Construye en el Sprint 0 la arquitectura de forma iterativa mediante un análisis preliminar de los conductores arquitectónicos (requisitos funcionales, de calidad y del negocio), y de un estudio de factibilidad del proyecto. El proceso de análisis de requisitos será entonces influenciado por las preguntas generadas durante el diseño arquitectónico. El resultado del Sprint 0 es un documento inicial que explica la arquitectura mediante un proceso de descomposición basado en los atributos de calidad del software.

Pero sería importante desarrollar un modelo genérico de estructura de software que pueda ser integrada a cualquier metodología ágil desde las fases iniciales de desarrollo y que incluya flexibilidad ante los cambios y adapten los criterios agilistas. Esto conlleva a decidir el paradigma más efectivo para integrar la arquitectura a los métodos ágiles, en el desarrollo de SI, respetando los criterios agilistas y que se distingan claramente los beneficios de aplicarla en todas las etapas de desarrollo.

Los autores del presente trabajo están elaborando un modelo que desde las MA se puede implantar una arquitectura que sea ágil también, basada en herramientas, patrones y

componentes reutilizables que tenga en cuenta los requisitos funcionales, no funcionales y atributos de calidad.

## Líneas de investigación y Desarrollo

Es de profundo interés para este grupo de investigación trabajar con la metodología ágil ICONIX Y SCRUM para lograr la integración de una Arquitectura de software que sea soportadas por ellas y no disminuyan sus capacidad, principios y criterios agilistas.

Los posibles beneficios de una integración de AS en las MA no es un tema lo suficientemente explorado en ambientes académicos, habiéndose encontrado escasa cantidad de publicaciones relacionadas.

Por otro lado, el interés de indagar en posibles estrategias de integración y posterior evaluación de riesgos/costos, surgieron de inquietudes emergentes en proyectos de investigación anteriores que este grupo llevó a cabo en [3 y 9], siempre siguiendo la línea de las Metodologías de desarrollo, en especial de las ágiles.

ICONIX: es una metodología ágil con un claro proceso simplificado, iterativo e incremental que unifica un conjunto de métodos de orientación a objetos con el propósito de abarcar todo el ciclo de vida de un proyecto, está adaptada a los patrones de UML, su proceso dirigido por casos de uso. Posee cuatro fases y cuatro hitos de fácil comprensión y aplicación. Cada paso que se realiza está definido por un requisito, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producido.

SCRUM: Sigue el desarrollo de las MA, parte de la visión, del concepto general del producto, y sobre ella el equipo produce de forma continua incrementos en la dirección apuntada por la visión; y en el orden de prioridad que necesita el negocio del cliente.

Comprende cinco fases y se construye el producto de forma incremental a través de

Información, tendiendo a lograr un equilibrio en la integración y reducir la exposición al riesgo.

iteraciones breves Estas iteraciones (en Scrum llamadas sprints) se repiten de forma continua hasta que el cliente da por cerrado el producto. Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución en reuniones breves diarias donde todo el equipo revisa el trabajo realizado el día anterior y el previsto para el siguiente. [3, 9].

En cuanto al análisis de las variables deseables que deben poseer las metodologías de software, las dos presentan virtudes destacables, Scrum es una que las favoritas por poseer certificaciones. Con relación a la evolución de un sistema después de mucho tiempo, Iconix es la que posee documentación suficiente para cumplir con esta condición.

Las dos metodologías cumplen casi sin diferencias entre ellas, el hecho de que deben cubrir el ciclo entero de desarrollo de software, integrar las distintas fases del ciclo de desarrollo, poder emplearse en un entorno variables de proyectos Software

## Resultados y Objetivos

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

- Proponer un modelo de Arquitectura de Software sin contraponerse con las postulaciones del manifiesto de las metodologías Ágiles.
- Aplicar la integración de Metodología Ágil y la Arquitectura de Software propuesta al desarrollo de diferentes SI de pequeñas organizaciones.
- Creación de un marco de referencia que permita analizar riesgos y estimar costos de la implementación de SI guiada por la integración propuesta.

El objetivo principal es establecer la incorporación de la AS al proceso de desarrollo dirigido por MA, con el propósito de que esa integración, respete los criterios, valores y principios del dominio de aplicación agilista, favorezca el diseño de Sistemas de

## Formación de Recursos Humanos

El equipo de trabajo está compuesto por cinco docentes-investigadores, y seis alumnos adscriptos. En el periodo 2016- 2017 se han asesorado: una tesis de maestría aprobada y finalizada, una tesis de maestría en proceso, una tesis de grado finalizadas y cuatro tesis de grado en proceso, tres de las cuales se están desarrollando en el nuevo tópico de investigación que se presenta en este trabajo.

Además, se espera realizar direcciones, a los alumnos adscriptos, en dos trabajos orientados al desarrollo de SI con metodologías ICONIX-SCRUM, adoptando el enfoque de integración con Arquitectura de Software.

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] Canos J., Letelier P. “Metodologías Ágiles en el Desarrollo de Software”. JISBD 2003. España
- [2] Navarro, Moreno, Aranda, Parra, Rueda “Integración de Metodologías Ágiles y Arquitecturas de Software en el desarrollo de Sistemas de Información” XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016) Entre Ríos, Argentina) ISBN: 978-950-698-377-2.
- [3] Navarro, Moreno, Aranda, Parra, Rueda. Proyecto de Investigación “Aplicabilidad de Metodologías y Tecnologías Informáticas en el Desarrollo de Sistemas de Información” Código: 21/E 979 –FCEFN- UNSJ – CICTCA- SIGEVA 2014-2015.
- [4] 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.
- [5] Manifiesto for Agile Software Development Kent Beck, Mike Beedle, et al. [www.agilemanifesto.org/](http://www.agilemanifesto.org/) Accedido el 22 de Febrero de 2016.
- [6] Breivold,H.P., Sundmark, D., Wallin, P. and Larsson, S., “What Does Research Say about Agile and Architecture?”, en Proceedings of the Fifth International Conference on Software Engineering Advances (ICSEA), USA, (2010).
- [7] ISO/IEC/IEEE42010. IEEE Std 1471:2000, “Recommended Practice for Architectural Description of Software intensive Systems”.
- [8] Martin Fowler. “Patterns of Enterprise Application Architecture” Addison- Wesley. 2003 1st Edition. ISBN-13: 007-6092019909.
- [9] Navarro- Rueda- Moreno – Aranda- Parra- “Convergencia de Tecnologías Informáticas y Metodologías para la Implementación de Sistemas de Información” Cod: 21/E/871 FCEFN.. 2012-2013. – CICTCA- SIGEVA.
- [10] Scrum y XP desde las trincheras - proyectalis.com Una historia de guerra Ágil SCRUM Y XP DESDE LAS TRINCHERAS Cómo hacemos Scrum Henrik Kniberg Prólogos de Jeff Sutherland y Mike Cohn ISBN: 978-1-4303-2264-1
- [11] Edwin Rafael Mago - Germán Harvey Alférez “El Papel de la Arquitectura de software en Scrum “ Publicado en SG #30 <https://sg.com.mx/revista/30/el-papel-la-arquitectura-software-scrum#.WNiAUtLhB1s>.