



Enterprise architecture and agile methodologies - An effective combination to tackle the frequent business changes

Arquitectura empresarial y metodologías ágiles - Una combinación efectiva para hacer frente a los frecuentes cambios en el negocio

Alejandro Álvarez C.¹, Albeiro Patiño B.²

¹[aalvarezc\(AT\)ucne.du.co](mailto:aalvarezc(AT)ucne.du.co), ²[ajpatinob\(AT\)ucn.edu.co](mailto:ajpatinob(AT)ucn.edu.co)
Fundación Universitaria Católica del Norte - Colombia

Artículo de reflexión

ABSTRACT

Businesses today are seeking major changes through a more agile approach to traditional models. Apparently, there is a trend already in place that leads to a transformation in the way enterprise architecture is performed by business. Traditional enterprise architecture development methodologies such as TOGAF, ZACHMAN, FEAF, among others, have been the main guide for projects that involve the design and deployment of software solutions. However, agile methodologies represent an opportunity to give value to the organization through collaborative, iterative and incremental work supported in the flowering and massive use of its models and techniques. This article attempts to explain how they have evolved from traditional enterprise architecture methodologies and the value it offers in conjunction with agile methodologies in IT projects.

Keywords: Agile architecture frameworks, agile methodologies, agile development frameworks, enterprise architecture, IT architecture.

RESUMEN

Actualmente las empresas buscan grandes transformaciones a través de un enfoque mucho más ágil a los modelos tradicionales. Al parecer está en formación una nueva tendencia que conlleva a una transformación en la forma como se aborda la práctica de Arquitectura empresarial. Las metodologías tradicionales de desarrollo arquitectura empresarial como TOGAF, ZACHMAN, FEAF, entre otras, han sido la guía principal para proyectos que tienen involucrado el diseño y construcción de soluciones de software. Sin embargo, las metodologías ágiles representan una oportunidad de dar valor a la organización a través del trabajo colaborativo, iterativo e incremental apoyado en el florecimiento y uso masivo de sus modelos y técnicas. El presente artículo intenta explicar cómo han evolucionado las metodologías tradicionales de arquitectura empresarial y el valor que ofrece en la conjunción con metodologías ágiles en los proyectos de TI.

Palabras clave: Metodologías ágiles de arquitectura, metodologías ágiles, metodologías ágiles de desarrollo de software, ágiles arquitectura empresarial, arquitectura TI.

© 2015. IAI All rights reserved

1 Introducción

La Globalización, los Tratados de Libre Comercio y la fuerte competencia que esto implica, demandan de las empresas modelos de desarrollo de software que les permitan ser rápidas y efectivas al tiempo que eficientes y oportunas. Refiriéndose a América Latina la Organización para la Cooperación y el Desarrollo Económico OCDE afirma que la innovación tecnológica resulta clave para promover el cambio estructural necesario, que favorezca la diversificación, incremente la productividad y reduzca las brechas tecnológicas en la región [5]. Una responsabilidad que no se le puede dejar solamente a las áreas de desarrollo, sino que debe partir, en primera instancia, de las áreas de arquitectura donde se deben dar los lineamientos para que las actividades de tecnología se puedan llevar a cabo de forma clara y ágil.

En procura de definir un modelo maduro que les permita a las áreas de tecnología aumentar la productividad de sus desarrollos y disminuir los costos y recursos (técnicos y humanos), requeridos para

optimizar el *time to market*, la Arquitectura Empresarial y las metodologías ágiles se constituyen en importantes pilares. La primera, como promotora de un *framework* o metodología que guíe la actuación de las personas y las áreas, y las segundas, como método de trabajo.

La Arquitectura Empresarial propone la creación de un modelo que permita abordar la tecnología como un complemento y apoyo del negocio, de forma que las áreas de Negocio y de TI evolucionen de manera alineada, conjunta, dinámica y ágil. De acuerdo con Bernhard Hitpaas [1], la tendencia de toda organización es crecer y desarrollarse, y el desarrollo es un proceso lento y gradual que conduce al conocimiento exacto de sí misma y a la plena realización de sus potencialidades. Por eso es que la tecnología debe cumplir su papel de evolucionar al mismo ritmo que el negocio impone, permitiendo un modelo que de forma coordinada plasme la realidad presente de la compañía y modele su visión futura. La evolución del modelo debe ser continua y sostenible, aplicable y fácil de implementar, soportar, actualizar y

mantener, sin que para ello se deba incurrir en costos elevados. Por su parte, desde su concepción, las metodologías ágiles surgieron como alternativa para subsanar los tantos y tan variados inconvenientes que enfrentan los desarrolladores al usar metodologías tradicionales. Pablo Lledó [3] relaciona algunas de ellas: falta de información, cambio de prioridades, esperas en colas, apagar incendios, falta de recursos, multi-tareas.

En este trabajo se abordan ambos temas: la Arquitectura Empresarial y las metodologías ágiles, y se muestra que su conjunción representa una combinación efectiva para hacerle frente a los frecuentes cambios de negocio. Se abordan algunas metodologías ágiles de Arquitectura en proceso de maduración y que están siendo usadas, que podrían servir para quienes planean diseñar y desarrollar software, con miras a mejorar la oportunidad tecnológica y el *time to market* de la empresa. Si bien es aplicable tanto a empresas grandes como medianas y pequeñas, podría ser más útil en las grandes y en algunos casos en las medianas, que es donde las metodologías tradicionales se vuelven más burocráticas y paquidérmicas.

Para la discusión y el análisis de los temas planteados aquí, los autores se basan en buena medida en su experiencia real y práctica, así como en una amplia bibliografía que se ofrece. Además, puede servir para profundizar en los conceptos planteados. Para llegar a la lista final de la bibliografía seleccionada, se realizaron búsquedas con base en expresiones claves, tales como Metodologías de Desarrollo de Software, Arquitectura Empresarial, *Framework* de Arquitectura, *Framework* de Desarrollo, entre otras. También se recurrió a la lectura y el análisis de revistas, libros e informes de organismos multilaterales, tales como la OCDE, el Banco Mundial, el Fondo Monetario Internacional, y otros tantos, toda vez que el propósito fundamental es proponer herramientas que ayuden a las empresas a aumentar la productividad, reducir los costos y mejorar el *time to market*, con un enfoque ágil.

Aunque los temas de Arquitectura Empresarial y metodologías ágiles datan de hace más de diez años, se procuró que la documentación para soportar el análisis fuera lo más reciente posible (de preferencia posterior a 2012), considerando que, particularmente, la tecnología sigue una línea acelerada en la que lo vanguardista pasa rápidamente a ser caduco. A continuación se revisan los siguientes temas: qué es un *framework* de arquitectura, para qué sirve y cómo se puede aplicar, qué es un modelo actual, un modelo transitorio y un modelo objetivo de arquitectura empresarial, que es un GAP, cuál es el rol de un arquitecto en proyectos ágiles y cuál ha sido el enfoque de las metodologías ágiles con mejores resultados en la práctica para que los equipos de trabajo realicen sus análisis y diseños de arquitectura empresarial.

2 Metodologías de Arquitectura Empresarial

En la Figura 1 se muestran algunas de las razones por las cuales las arquitecturas se vuelven caóticas y difíciles de administrar en las empresas.

Gestionar y administrar una arquitectura regida por este tipo de antecedentes y premisas de crecimiento es

difícil y complejo. De ahí que se haga necesario definir un mecanismo de gestión y administración, y en este punto es donde surge la Arquitectura Empresarial. Institute of Electrical and Electronics Engineers IEEE ratificó en 2000 el estándar 1471-2000 como una práctica recomendada para la descripción de sistemas intensivos de software, creando un modelo conceptual para la descripción de metodologías específicas de arquitectura, e incluyendo una aproximación fiable al *framework* de arquitectura empresarial. Este estándar sirvió como fundamentación conceptual para entender y modelar los puntos de vista individuales y las perspectivas y preocupaciones de los usuarios de negocio.



Figura 1: Razones por las que se complejizan las arquitecturas en las empresas (A partir de [4])

Por su parte, The Open Group Architecture Framework TOGAF [2] define la Arquitectura Empresarial como la organización fundamental de un sistema, conformada por sus componentes, las relaciones entre ellos y su entorno y los principios que gobiernan su diseño y evolución. TOGAF propone trabajar con cuatro arquitecturas:

1. Arquitectura de negocio, se ocupa de la estrategia de negocio, gobierno, organización y procesos clave de la compañía.
2. Arquitectura de datos, es la responsable de definir la estructura de datos lógicos y físicos que posee una organización y sus recursos de gestión de datos.
3. Arquitectura de aplicación, debe atender todo lo concerniente a las aplicaciones individuales que se deben implementar, sus interacciones y las relaciones que tienen con los procesos de negocio.
4. Arquitectura tecnológica, debe garantizar las capacidades software y hardware para apoyar la implementación de los servicios de negocio, de datos y de aplicación.

Siguiendo con TOGAF, el *framework* de Arquitectura Empresarial proporciona un modelo detallado de productos de trabajo arquitectónicos, incluyendo entregables, artefactos dentro de los entregables y los Bloques de Construcción de la Arquitectura (ABB) que los entregables representan.

En cierta forma, Zachman [8] fue quien acuñó el concepto de Arquitectura Empresarial al que se refirió como Sistema de Información en su trabajo. Pero ante todo su propósito era sentar las bases para alinear la tecnología y el negocio, y procurar lograr mejores resultados en tiempo, costos, recursos y *time to market* para la compañía. De esta forma Zachman creó la punta de lanza para abrir el camino. De sus raíces crecieron otras visiones de Sistemas de Información y Arquitectura Empresarial. El Departamento de Defensa de los Estados Unidos creó en 1994 el Technical Architecture

Framework for Information Management (TAFIM); posteriormente, en 1996, evolucionó al Federal Enterprise Architecture Framework (FEAF), que nuevamente en 2002 cambia el nombre por el actual Federal Enterprise Architecture (FEA). De esta base nacen los trabajos de The Open Group TOGAF.

Quizá los *frameworks* de Zachman y TOGAF sean los más conocidos y utilizados a nivel mundial, tanto en el sector público como en el privado. Sin embargo, y a partir de ellos, muchas compañías han definido los suyos buscando siempre mantenerse dentro de los lineamientos generales dictados por los originales. También particularizan los suyos de forma que se ajusten a sus necesidades y a sus formas particulares de ver y aplicar la alineación de negocio y tecnológica. En la Figura 2 se muestra un *framework* definido para una empresa particular, que considera que la mejor forma de alinear negocio y tecnología es ver a la Arquitectura Empresarial con especial atención a las aristas de: 1) procesos, 2) información, 3) aplicativa, 4) integración, 5) técnica, y 6) de seguridad.



Figura 2: Framework de Arquitectura Empresarial para una particular empresa (Autor)

2.1 Utilidad de un *framework* de Arquitectura Empresarial

La Arquitectura Empresarial es una representación gráfica en la que se muestran diagramas y especificaciones teóricas, que facilitan la comprensión de los procesos de negocio, así como la información, los Sistemas de Información, los servicios que permiten sus interacciones, etc. De forma que se puedan optimizar los recursos (técnicos y humanos), mantener un adecuado balance de la relación costo/beneficio y tener satisfechos a los usuarios. Una de las principales ventajas de la Arquitectura Empresarial es que facilita la adaptabilidad del negocio, y como afirman Sáez et al. [7], los Sistemas Adaptativos Complejos CAS pueden reorganizar su estructura interna sin la intervención de ningún agente externo. Esta característica, producto un aprendizaje inconsciente, aumenta la probabilidad de que el sistema sobreviva en entornos turbulentos e inestables. Otras ventajas de tener claramente definida la Arquitectura Empresarial son:

- Los procesos se identifican y mantienen más fácilmente.
- Se garantiza de forma estructurada una permanente alineación entre el negocio y la tecnología.
- Se tiene identificados claramente los componentes tecnológicos que se usarán para soportar los procesos de negocio.
- Se hace más fácil la toma de decisiones en relación con la tecnología y en función de los procesos del negocio.

- Se pueden tomar decisiones más fáciles y rápidamente, con oportunidad y garantizando una buena relación costo/beneficio a través del reuso de activos de TI (aplicaciones, datos e infraestructura)
- Se hace más fácil identificar dónde están los focos de interés y de mayor oportunidad, y los que presentan mayores riesgos y preocupación, lo que sirve como guía para la evolución del negocio a corto, mediano y largo plazo.
- Es más fácil determinar los impactos que se visualizan ante cambios en los procesos de negocio y en la tecnología.

2.2 Modelo actual, modelo transitorio y modelo objetivo de la Arquitectura Empresarial

De acuerdo con Ross, Weill y Robertson [6], la Arquitectura Empresarial, como conjunto integrado de procesos de negocio, puede mejorar los servicios para los clientes, suministrar mejor información para la toma de decisiones y permitir cambios en una parte de la empresa para alertar a otras sobre acciones que deben tomar. De lo anterior se desprende que hay diversos aspectos críticos que atañen a la Arquitectura Empresarial, tales como los procesos de negocio y los servicios para los clientes, la información y los cambios que impactan las decisiones que se deben tomar. Estos últimos, entre todos, son quizás los que más relevancia cobran cuando se trata de administrar una Arquitectura Empresarial, por lo que mantenerla actualizada es vital para el cumplimiento de los demás aspectos. Los cambios deben reflejarse en los modelos de arquitectura y si no sucede así, lo más seguro es que la Arquitectura misma fallará como modelo de alineación de negocio y tecnología.

Ross et al. [6] proponen que el mejor soporte para la estrategia de una compañía es el modelo que defina. Pero como tanto el negocio como la tecnología cambian permanentemente, es necesario garantizar una evolución sistemática en los modelos. No es fácil hacer un cambio radical en la Arquitectura Empresarial, pero de hacerlo es posible que se impacte negativamente al negocio por posibles fallas en la tecnología. Lo ideal es definir un *roadmap*, en el que se pueda visualizar claramente el hoy, el mañana cercano y el futuro a mediano plazo, tanto en los modelos de negocio como en los de Arquitectura, o lo que es lo mismo, en la Arquitectura Empresarial de la compañía. En otras palabras, la Arquitectura Empresarial debe procurar contar con diseños que reflejen la situación actual de la empresa y con diseños que muestren la situación que visualiza para su futuro mediano. Pero, además debe diseñar un modelo que permita, de forma paulatina y gradual, ir acercándose a ese modelo objetivo. En la Figura 3 se representa el *roadmap* por el que debe propender la Arquitectura Empresarial para garantizar una efectiva evolución desde un modelo actual a un modelo objetivo.

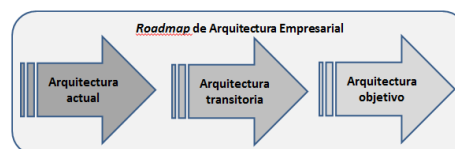


Figura 3: Roadmap de evolución de una Arquitectura Empresarial (Autor)

2.3 Brechas entre los modelos de Arquitectura Empresarial

Para progresar desde el modelo actual al modelo objetivo de la Arquitectura Empresarial se debe evidenciar las falencias y vacíos, es decir, las brechas o GAPS que se deben cubrir. Estos GAPS pueden deberse a aspectos puntuales en cualquiera de las vistas de la arquitectura: la técnica, la aplicativa, la de integración, la de información o la de seguridad. Las razones para que existan en la compañía pasan por la disponibilidad de recursos técnicos o humanos, falta de conocimiento, costos, oportunidad, etc., que se deben evidenciar y reflejar en la arquitectura transitoria, y definirse la forma y fecha para la cual se espera cubrir.

En cualquier caso, siempre se podrá considerar osado, sobre todo en proyectos grandes, intentar pasar de un modelo de arquitectura actual a un modelo de arquitectura objetivo sin definir un modelo transitorio que permita contar con todos los recursos técnicos y humanos requeridos, el personal debidamente capacitado, el presupuesto garantizado, etc. Algo que confirman Matsumura, Brauel y Shah [4] al recomendar que no se debe intentar una aproximación de *big bang*, es decir, hacer realidad el esquema global de la Arquitectura mediante un único, interminable y costosísimo proyecto. Lo mejor es seleccionar y establecer una secuencia de pequeños proyectos, de forma que cada uno aporte por sí mismo una ventaja cuantificable para el negocio.

Cubrir un GAP en la Arquitectura Empresarial podría ser la instalación de un canal de comunicación o una aplicación, o apropiar conocimiento en la gestión y administración de una determinada herramienta; también garantizar que una transmisión de archivos, que tradicionalmente se ha hecho de forma no-segura, se haga de forma segura, o cambiar el mecanismo de intercambio de información entre dos aplicaciones que típicamente han *hablado* mediante servicios web a un mecanismo basado en colas. Incluso, puede ser la conjunción de varios de estos u otros inconvenientes, o hasta el emprendimiento y finalización de uno o varios proyectos. En la Figura 4 se muestra la arquitectura actual y la objetivo, y entre ellas los GAPS a superar.

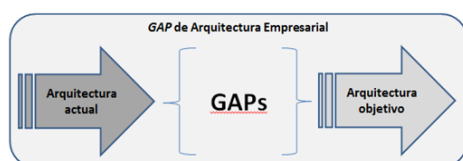


Figura 4: GAPS de la Arquitectura Empresarial (Autor)

Matsumura, Brauel y Shah [4] afirman que cada proyecto debe proporcionar un retorno de la inversión, y además motivar la realización de futuros proyectos que le permitan continuar surcando el espacio hacia sus objetivos. También dicen que a medida que se implementa se pueden ir perfeccionando y automatizando los procesos de implementación de la Arquitectura, hasta alcanzar una condición en la que los esfuerzos desaparecen, o estado de *ingravidez*.

2.4 Las metodologías ágiles sustentan la Arquitectura Empresarial

En este punto es donde cobran valor y relevancia las metodologías ágiles. En el proceso de sacar adelante uno,

o en el peor de los casos, varios proyectos de forma simultánea, como ruta crítica para pasar de una arquitectura actual a una arquitectura objetivo, puede ser necesario tomar muchas y muy variadas decisiones, y probablemente de forma apresurada. Si a esta dificultad se le suma el que es una empresa grande, donde la burocracia pesa y hace lentos los procesos, el asunto se agrava. En una situación como esta las metodologías tradicionales no son tan útiles, porque las actividades se realizan más en secuencia que en paralelo, es necesario involucrar a personas muchas veces solamente en virtud de su rango, o por escalamiento. En este caso, los procesos se hacen lentos, los reprocesos aumentan y las decisiones se retardan.

Las metodologías ágiles se presentan como una nueva apuesta para los desarrolladores de software, justamente en respuesta a las falencias de las metodologías tradicionales, incluso con mayor aceptación por parte de los líderes PMO. Esencialmente por ser simples y prácticas, por reunir en una misma mesa a todo el equipo de trabajo, ser más colaborativas y flexibles, y sobre todo por lograr resultados en menos tiempo. Una vez conocido el modelo actual y el objetivo de la Arquitectura Empresarial es necesario definir una forma controlada de avanzar. Lo mejor es hacerlo mediante iteraciones, es decir, de forma incremental, evolutiva, escenificada, o en espiral, porque no es algo nuevo. Lo único que se busca es un mecanismo de retroalimentación que permita saber con precisión cuál es la situación en intervalos cortos y frecuentes.

En la misma mesa de trabajo deben estar sentados, de preferencia, los integrantes de las diferentes áreas que conforman la cadena de desarrollo; y en conjunto, sin perder de vista ni desconocer la responsabilidad que recae sobre cada uno, serán los encargados y deben identificar: servicios de negocio, requisitos para la descripción de los servicios, métricas de rendimiento de los servicios, estándares de interoperabilidad, esquemas de datos, políticas, requisitos de clasificación y localización de los servicios [4]. Una vez identificados estos insumos el equipo debe trabajar en la resolución de los problemas planteados, documentarlos y proceder luego a su implementación.

2.5 Estado de las metodologías ágiles de Arquitectura Empresarial

Actualmente, las empresas buscan grandes transformaciones a través de un enfoque mucho más ágil de los modelos implementados. Parecería que se está formando una nueva tendencia que conlleva a una transformación en la forma como se aborda la práctica de la arquitectura empresarial. Esta tendencia está relacionada con conceptos actuales y relacionados entre sí, tales como metodologías ágiles, *cloud computing*, transformación digital, iteraciones continuas y desarrollo rápido de software. Todo esto está marcado por la necesidad de acelerar el negocio y en particular el ciclo de desarrollo y la salida al mercado de nuevos productos y servicios.

La arquitectura empresarial todavía es una práctica que pueden implementar solamente empresas grandes con capacidad de ejecución. Una de las principales razones para la no-adopción es que los tiempos de

implementación y maduración son prolongados y a corto plazo no se obtienen beneficios tangibles, por lo que variables como la incertidumbre en tiempos y costos únicamente pueden ser asumidas por grandes empresas con grandes presupuestos. La metodología TOGAF contiene un conjunto de guías y técnicas para desarrollar una arquitectura empresarial, y su desarrollo y puesta en marcha contempla ocho fases. Cada una requiere alrededor de 20 entregables, lo que necesita tiempos de implementación prolongados y una restricción de entrada en términos de agilidad en la adopción de la metodología para el desarrollo de una Arquitectura empresarial [9]. Sin embargo, la necesidad de responder rápidamente a los requerimientos del negocio y a las nuevas tendencias del mercado es una exigencia a la que todas las empresas están expuestas sin importar su tamaño, recursos o sector. Diversos marcos metodológicos generales y particulares están disponibles para cada sector industrial, y ofrecen una guía de implementación a alto nivel. Pero cuando se organiza una iniciativa en su desarrollo, siguiendo algunos de los modelos de implementación disponibles, los tiempos de puesta a punto e implementación y la cantidad de entregables y artefactos es una restricción para la mayoría de empresas, especialmente Pymes.

Las empresas y el mercado son cada vez más exigentes y demandan alta productividad, mayor flexibilidad a los constantes cambios, disminución de los ciclos de entrega de servicios de software y la implementación de capacidades de negocio que habiliten los nuevos productos y servicios [10]. La adopción de metodologías ágiles es una forma para lograr estándares de eficiencia y satisfacción del negocio. Es común encontrarlas aplicadas a distintas disciplinas, tales como el desarrollo de software bajo metodologías SCRUM, eXtreme Programming, Lean y Six Sigma, entre otras, que ofrecen un modelo para gestionar eficientemente el ciclo de desarrollo de software, con una cantidad de actividades alcanzable y tareas en contextos ágiles.

Para la Arquitectura Empresarial el concepto de agilidad no ha evolucionado ni ha sido incorporado en sus modelos de referencia, como si lo han hecho las disciplinas descritas. Sin embargo, existen percepciones divergentes en la relación de uso entre las metodologías ágiles y la arquitectura. Si bien el agilismo sirve para mejorar la agilidad en el desarrollo de proyectos, también ha sido criticado por practicantes y académicos debido a la falta de un enfoque en la práctica en la Arquitectura [17, 18]. De acuerdo con Beck [19], algunas metodologías ágiles, como eXtreme Programming (XP), sugieren la eliminación de artefactos de Arquitectura que no estén relacionados con la versión actual del sistema que se está tratando en el proyecto. Algunos autores también sugieren que las metodologías ágiles son demasiado ligeras en cuanto a cumplir con las necesidades arquitectónicas y no son abordadas con suficiente profundidad y rigurosidad [20].

Estudios documentados sobre la relación entre metodologías ágiles y Arquitectura muestran que apenas si la abordan, y sugieren hacerlo de forma más amplia y con nuevos métodos asociados directamente al trabajo arquitectónico, el cual es considerado como una de las tareas más importantes en el diseño inicial de los

proyectos [21, 24, 25]. A esto se suma la creciente necesidad de reconocer la importancia del buen diseño en las etapas iniciales del proceso de desarrollo de software, y de integrar métodos de arquitectura en los enfoques ágiles [22]. En el mismo sentido, para Falessi et al. [23] el trabajo de Arquitectura es relevante, especialmente en el entendimiento y modelamiento del sistema, la documentación de las alternativas o escenarios de implementación, la oportunidades en el escalamiento de las prácticas ágiles de forma transversal en otros proyectos y la reusabilidad de componentes para soportar futuros requerimientos.

2.6 El rol del arquitecto ágil

En la ejecución de proyectos ágiles se observa la necesidad de tener en el equipo un rol que haga las veces de *architect owner*, quien será el responsable de diseñar la arquitectura de una forma distinta y especializada para proyectos ágiles. Es to sería similar al rol actual de *product owner*, quien es responsable de la definición y validación de los requisitos, o al *scrum master* como responsable de implementarlos. El *architect owner* tiene un perfil técnico y es responsable de la definición y el modelamiento de la arquitectura, con una visión transversal que facilite su evolución en el corto, mediano y largo plazo con criterios de simplicidad y flexibilidad [24]. Tradicionalmente, el arquitecto se ha encargado exclusivamente de su definición y de verificar su cumplimiento. En el enfoque ágil, realiza su función de forma colaborativa con el equipo, que tiene una función de co-ayudar a su evolución en el tiempo. De forma similar, el rol de arquitectura puede ser transicional a otros miembros del equipo. El *architect owner* posee un rol senior, lo cual le facilita servir de mentor a sus pares. Uno de los retos que tiene bajo el esquema de los equipos ágiles es definir las guías de implementación de TI, bien sea de seguridad, información, integración, etc., con la agilidad necesaria que no estanque el normal desarrollo del equipo.

Si bien la arquitectura puede ser definida en consenso con el equipo, en algunas ocasiones no se llega a un acuerdo, y entonces el arquitecto debe ser quien tome decisiones con la oportunidad necesaria para no retrasar al equipo ni al proyecto. En la metodología tradicional de cascada el arquitecto construye las definiciones en las etapas iniciales del proyecto, particularmente en la fase de visión o planeación; pero en los proyectos ágiles tiene una característica natural, definida en el manifiesto ágil y asociada a los requisitos cambiantes entre un *sprint* y otro. Por eso el rol del arquitecto en proyectos ágiles es más intensivo. Usando estas metodologías, muchos cambios solicitados entre los *spints* pueden impactar la arquitectura y requerir nuevas definiciones, lo que la convierte en un rol con mayor interacción en la participación a lo largo del proyecto. El trabajo del arquitecto no solamente aparece al inicio sino también durante la ejecución del proyecto, particularmente después de la planeación del *sprint*.

En la Figura 5 se explica cómo se organiza el *architect owner* y otros miembros del equipo entorno a los requisitos definidos, y donde debe colaborar con los otros miembros con el objetivo de evolucionar la arquitectura de la solución. Generalmente, los

subsistemas tienen definidos sus propios *architect owners*, como responsables de visionar la arquitectura al inicio del proyecto y de coordinar la toma de decisiones arquitectónicas con el equipo y con el apoyo de asesores de otros dominios.

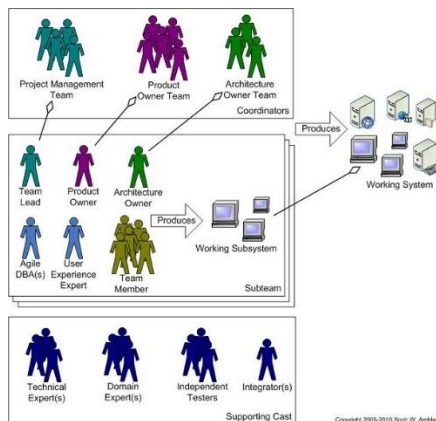


Figura 5: Estructura organizacional en proyectos ágiles [13]

Díaz et al [26] presentan otro enfoque (Figura 6) a través de un modelo adaptado de la metodología SCRUM, en el que el *product owner* define las características del producto y que son listadas en las historias de usuario, conocidas como *product backlog*. Posteriormente, estas historias son priorizadas de acuerdo con las necesidades del negocio y se asignan a las iteraciones o *sprints*. Cada una de las iteraciones comienza con una sesión de planeación, donde el *product owner* y el equipo planean lo que se debe hacer. En este modelo adaptado de SCRUM, el arquitecto ágil realiza su labor en conjunto con el equipo y de forma evolutiva e incremental. Este enfoque es flexible y los cambios se permiten a través de mecanismos de extensión. El modelo determina que el trabajo del arquitecto se enfoca en la toma de decisiones de diseño, dependencias, limitaciones y renuncias. Estas variables deben ser balanceadas con las prioridades del negocio. Durante la sesión de planeación del *sprint* se define los modelos de arquitectura que serán implementados. En la reunión de revisión del *sprint* el *product owner* evalúa el producto potencialmente entregable y el nivel de adherencia o cumplimiento de la arquitectura.

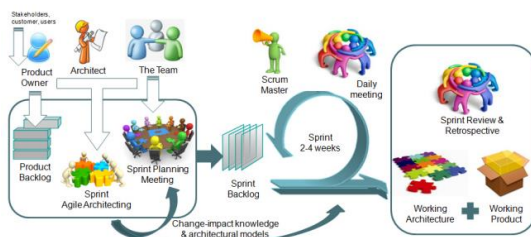


Figura 6: Modelo SCRUM adaptado con arquitectura ágil [26]

3 Aproximación a algunas metodologías ágiles de Arquitectura Empresarial

Las metodologías ágiles fueron concebidas para procesos de desarrollo de software y enfatizan en un desarrollo rápido y flexible. La definición de las arquitecturas, en alguno de sus campos de acción (aplicaciones, información, tecnológica, de integración o de seguridad) es una tarea importante en la planeación y ejecución de los proyectos ágiles. La combinación de metodologías se hace necesaria para lograr la agilidad deseada. Unir los beneficios de estas metodologías con

las tradicionales es una opción que ha tomado fuerza en la transición a nuevos modelos de gestión tecnológica.

3.1 Arquitectura de dos velocidades

La habilidad para desarrollar y entregar nuevos productos de forma oportuna se ha convertido en un factor competitivo para las empresas. Uno de los conceptos que está surgiendo para administrar la complejidad y agilizar los ciclos de entrega de productos es la Arquitectura de dos velocidades, o *Two-speed architecture*. Este tipo de arquitectura requiere una ejecución rápida de cara al cliente, que se conjuga con una arquitectura tradicional centrada en el *legacy* [11]. Para lograrlo se requiere componentes modulares y altamente desacoplados, para permitir el despliegue rápido de nuevos programas o servicios y evitando en mayor medida el largo trabajo de integración. Adicionalmente, se debe definir y construir micro-servicios (pequeñas funcionalidades desacopladas), que estarán disponibles en los canales de interacción con el cliente y con un tiempo de implementación reducido, generalmente días, a diferencia de los servicios rutinariamente concebidos. Todo esto debe estar acompañado de cero tiempos de indisponibilidad, que afectan la confianza y experiencia del cliente, además de analítica en tiempo real, alta capacidad de configuración y reusabilidad de los componentes. Sin embargo, crear tal capacidad empresarial no necesariamente es fácil y requiere de la conformación de nuevas herramientas y la modificación de procesos de gobierno de TI [27].

Estas capacidades pueden ser naturales para las empresas modernas, pero las establecidas y con larga permanencia en el mercado pueden encontrar dificultad a la hora de desarrollar las capacidades digitales, debido a que todavía tienen procesos de negocio soportados en sistemas *legacy* en el *back end*. Estas empresas tienen una alta presión para renovar sus arquitecturas a través de proyectos de transformación de corta duración. Se hace necesario entonces crear un modelo de gobierno que facilite la interacción entre TI y el negocio de manera clara y efectiva, para garantizar la transformación a la arquitectura digital. El modelo de arquitectura de dos velocidades también se enfrenta a retos, principalmente en la transformación de sistemas legados, que generalmente tienen mayor complejidad a la hora de evolucionarlos haciendo que los proyectos de transformación tengan mayor costo y duración.

3.2 Lean architecture

El concepto *lean architecture* nace de la aplicación de los principios del sistema de producción de Toyota y mantiene los mandatos de la metodología *lean* en cuanto a deshacerse de lo que no genera valor intrínseco. Esta arquitectura se relaciona con un concepto de arquitectura ligera combinada con la consistencia y la reducción del desperdicio. El modelo ofrece el concepto de que la arquitectura ágil debe ser definida de acuerdo con la experiencia del equipo, lo que implica que el sistema puede ser mejorado y evolucionado día a día en la medida en que los requisitos se documentan y el equipo mejor la comprensión y claridad de los mismos [16]. Se trata de un enfoque incremental dirigido por la experiencia y el conocimiento del equipo y que honra el

valor de la adaptación. A medida que el equipo madura en el entendimiento de los requerimientos, la arquitectura madura a su ritmo.

El modelo *lean architecture* también adopta la perspectiva de que la documentación innecesaria es un factor que genera alto costo, y en la mayoría de casos es innecesaria y termina archivada y sin uso. También enfatiza en la comunicación a través de medios eficientes y en la discusión cara a cara. Esta metodología critica y evita la excesiva planeación con enfoques a largo plazo, centrándose en lo que es importante ahora e inmediato para tomar decisiones y producir productos de forma inmediata, proporcionando valor verificable y comprobable a los usuarios del negocio.

3.3 Scaled Agile Framework SAF

Esta metodología es una práctica asociada al modelo *lean* y está enfocada principalmente en proveer un conjunto de patrones que puedan ser usados para construir nuevas soluciones software, y evolucionar las existentes de forma rápida. Históricamente, las metodologías SCRUM, XP y otras, se han enfocado en el equipo y en el proceso de desarrollo. Mientras que SAF presenta una vista ejecutiva y unificada que permite subir de nivel para análisis y tendencias o para desagregar en mayor detalle para la toma de decisiones [14]. Generalmente, un equipo de trabajo bajo la metodología SAF se compone de entre 8 y 10 miembros, que tienen una visión *end-to-end* de los requerimientos. También existen proyectos que pueden contener varios equipos con tamaños diversos y con integrantes de distintos perfiles que tienen el objetivo de sincronizar y alinear los *sprint's* y las entregas, de forma tal que la labor de codificación se puede realizar de forma paralela. Estos equipos se reúnen al comienzo de cada ciclo de salidas o liberaciones para definir los incrementos de cada *sprint*. En SAF, para asegurar que el desarrollo se hace bien se sugiere un conjunto de prácticas dirigidas más hacia la prevención que a las pruebas. El modelo comienza con un diseño de arquitectura ágil considerado como emergente o ascendente, y que evoluciona de forma colaborativa. Esto parte de uno de los principios del manifiesto ágil, en el que el mejor diseño de arquitectura nace de equipos que son autónomos [15]. Uno de principios bajo el modelo SAF es construir la arquitectura más simple posible, que permita el funcionamiento del sistema con los requerimientos definidos ya incorporados. Esto soporta uno los principios del manifiesto ágil que aduce que son bienvenidos los requisitos cambiantes incluso cuando se ha iniciado a labor de codificación [15].

4 Conclusiones

A pesar que las metodologías ágiles de desarrollo de software han recibido atención durante los últimos años, ha habido percepciones divergentes sobre la relación entre métodos ágiles y arquitectura. Diversos autores y estudios han criticado el rol de la arquitectura en los proyectos de desarrollo de software, por el retraso que generan y la posibilidad de crear una arquitectura a medida que el proyecto evoluciona. Sin embargo, si en las etapas previas no se cumplen las condiciones de diseño establecidas por la arquitectura, las correcciones en las etapas posteriores son costosas [25].

La mayoría de *framework* o metodologías tradicionales de Arquitectura se pueden usar en ambientes ágiles, porque ofrecen un enfoque incremental e iterativo. Pero las nuevas metodologías enfocadas en agilidad ofrecen métodos más claros para su desarrollo [12], porque están enfocadas en un diseño incremental y colaborativo de la Arquitectura que debe proporcionar los requisitos mínimos para soportar los requerimientos del negocio, y posteriormente evolucionar a un estado deseado de la arquitectura. El uso de las metodologías ágiles en el trabajo de la Arquitectura se ha desplegado de forma masiva por sus ventajas en términos de acortar los tiempos de entrega de soluciones que soportan la evolución de las empresas. Sin embargo, es necesario modificar los procesos de gobierno de TI para que soporten su uso.

Referencias

- [1] Hitpass, B. (2012). [Business Process Management \(BPM\). Fundamentos y conceptos de implementación](#). Santiago: Editorial BHH Ltda.
- [2] Josey, A. et al. (2013). [TOGAF versión 9.1 - Guía de Bolsillo](#). Zaltbommel: Van haren Publishing.
- [3] Lledó, P. (2014). [Gestión Lean y ágil de proyectos. Lean Project Management](#). USA: PabloLledó Ediciones.
- [4] Matsumura, M. et al. (2009). [Adopción de SOA para Dummies®](#). New York: Wiley Publishing Inc.
- [5] OCDE (2013). [Perspectivas económicas de América Latina 2014 - Logística y competitividad para el desarrollo](#). México: OCE - CEPAL.
- [6] Ross, J.; Weill, P. & Robertson, V. (2006). [Enterprise Architecture as strategy: Creating a foundation for business execution](#). USA: Harvard Business School Press.
- [7] Sáez, V. et al. (2003). [Temas básicos de innovación tecnológica en las empresas](#). Madrid: Universidad Politécnica de Madrid.
- [8] Zachman, J. (1987). [A framework for Information Systems](#). The IBM Systems Journal 26(3), pp. 454-470.
- [9] González, S. (2013). [Enterprise Architecture and Organizational Transformation](#). Proceedings Delivering Business Value through Enterprise Transformation, pp. 12-19. San José, Costa Rica.
- [10] Glaiel, F. (2012). [Agile project dynamics: A Strategic project management approach to the study of large-scale software development using system dynamics](#). Massachusetts: MIT Sloan School of Management.
- [11] Bossert, O., Ip, C. & Laartz, J. (2014). [A two-speed IT architecture for the digital enterprise](#). USA: McKinsey Quarterly Report.
- [12] Coplien, J. (2012). [The architecture owner role: How architects fit in on agile teams](#). Online [Mar. 2015].
- [13] Heusser, M. (2015). [Introducing the scaled agile framework](#). CIO Magazin. Online [Mar. 2015].
- [14] Cunningham, W. (2001). [Manifiesto for agile software development](#). Online [Feb. 2015].
- [15] Coplien, J. & Bjørnvig, G. (2010). [Lean Architecture for agile software development](#). West Sussex: Wiley.
- [16] Dybå, T. & Dingsøyr, T. (2008). [Empirical studies of agile software development: A systematic review](#). Information and Software Technology 50(9-10), pp 833-859.
- [17] Hochstein, L. & Lindvall, M. (2005). [Combating architectural degeneration: A survey](#). Information and Software Technology 47(10), pp 643-656.
- [18] Beck, K. & Andres, C. (2004). [Extreme Programming Explained: Embrace Change](#). Boston: Wesley.
- [19] Breivold, H. et al. (2010). [What does research say about agile and architecture?](#) Proceedings Fifth International Conference on Software Engineering Advances, pp 32-37. Nice, France.

- [20] Baba, M., & Abrahamsson, P. (2008). [Architecture-Centric Methods and Agile Approaches](#). Lecture Notes in Business Information Processing 9, pp 242-243.
- [21] Duka, D. (2013). [Adoption of agile methodology in software development](#). Proceedings International Convention on Information & Communication Technology Electronics & Microelectronics, pp. 426-430. Opatija, Croatia.
- [22] Falessi, D. et al. (2010). [Peaceful Coexistence: Agile Developer Perspectives on Software Architecture](#). IEEE Software 27(2), pp. 23-25.
- [23] Booch, G. (2009). [The Defenestration of Superfluous Architectural Accoutrements](#). IEEE Software 26(4), pp. 7-8.
- [24] Kruchten, P. (2009). [Software Architecture and Agile Software Development - An Oxymoron?](#) Workshop Keynote software architecture challenges in the 21st century. Los Angeles.
- [25] Díaz, J. et al. (2013). [Change-Impact driven Agile Architecting](#). Proceedings 46th Hawaii International Conference on System Sciences, pp 4780-4789.
- [26] Golden, B. (2015). [What Gartner's Bimodal IT Model Means to Enterprise CIOs](#). CIO. Online [Feb. 2015].