

Integración de ADM y Métodos de Desarrollo de Software

ADM Integration and software development methods

Fredy Gutiérrez

Ingeniero de
Sistemas
Universidad
Nacional.
Especialización en
Ingeniería de
Software
Universidad Distrital,
Colombia

faguitierrez@correo
.udistrital.edu.co.

Recibido enero 2013
Aceptado abril 2013

RESUMEN

Existe una amplia lista de metodologías, estándares y procesos en tecnologías de la información que sirven de guía al interesado para resolver o apoyar la solución de problemas, por ejemplo, ciclo de vida de desarrollo, calidad del producto, estandarización de procesos, gobierno de procesos, gestión de proyectos, arquitecturas de software, de sistemas o empresarial entre otras. El compendio es extenso y dinámico en la medida que año tras año se crean nuevas metodologías, guías y estándares que intentan mejorar la solución o resolver nuevos problemas. El marco de trabajo empresarial TOGAF posee una facilidad natural para absorber estándares de manera dinámica gracias a su alto nivel de abstracción, esta característica le permite incorporar a su repositorio artefactos construidos a partir de diferentes estándares o métodos de la industria, por ejemplo Cobit o Itil para gobernabilidad y gestión de infraestructura o inclusive entregables construidos en otros marcos de trabajo como Zachman. El presente artículo expone los enfoques que existen para articular las metodologías de desarrollo de software bien sea ágiles o disciplinadas con las fases de desarrollo del método ADM del marco de trabajo TOGAF.

Palabras clave— marcos de trabajo, metodologías, estándares, artefactos

ABSTRACT

standards and processes in information technology which guide the interested to solve or support the solution of problems, for example, development life cycle, product quality, process standardization, process governance , project management, software architecture, business systems or among others. The compendium is extensive and dynamic average year after year creating new methodologies, guidelines and standards that try to improve the solution or solve new problems. The business framework TOGAF has a natural ease to absorb dynamically standards thanks to its high level of abstraction, this feature allows you to add to your repository artifacts constructed from different standards or methods such as industry or Itil Cobit for governance and infrastructure management or deliverables including built in other frameworks such as Zachman. This paper presents the approaches that exist to incorporate the software development methodologies either agile or disciplined with the development phases of the ADM method TOGAF framework.

Key Word — frameworks, methodologies, standards, artifact

INTRODUCCIÓN

Como parte del proyecto de grado “Modelo de una plataforma interoperable para el gobierno Nacional” se exploró la integración del método de desarrollo ADM para arquitecturas empresariales con los diferentes métodos de desarrollo de software.

El método ADM y en general el marco TOGAF realiza el análisis arquitectónico con alto nivel de abstracción para visualizar, detectar y documentar oportunidades y riesgos durante el desarrollo de la arquitectura, esta ventaja cambia al momento de seleccionar el método de desarrollo puesto que el marco TOGAF es agnóstico a esas metodologías y permite total discrecionalidad al arquitecto empresarial y al arquitecto de software en la selección del método.

Los numerales II y III del artículo presentan una breve descripción de ADM y de los métodos de desarrollo actuales, posteriormente se exponen los diferentes enfoques de integración de los métodos para luego finalizar con algunos conceptos obtenidos durante la investigación.

METODO DE DESARROLLO DE LA ARQUITECTURA ADM

El marco de trabajo (framework) TOGAF consiste en un método detallado y un conjunto de herramientas que direccionan la arquitectura empresarial. Está compuesto por cuatro partes principales: el método de desarrollo de la arquitectura ADM (Architecture Development Method), la taxonomía empresarial (Enterprise Continuum) y la base de recursos (Architecture Repository).

TOGAF aborda el desarrollo a partir de cuatro niveles de abstracción: arquitectura de negocio, arquitectura de sistemas de información, arquitectura de datos y arquitectura tecnológica. ADM refleja estos niveles de abstracción en diferentes fases que determinan la línea base (baseline o as-is) y final de un nivel de abstracción (target o to-be)

junto con un análisis de brecha (gap analysis) que permite conocer el estado final de la arquitectura después de una o varias iteraciones.[1]

El método define ocho niveles A, B, C, D, E, F, G, H que van desde la visión de la arquitectura hasta la administración del cambio, por ser una metodología iterativa permite completar, eliminar o crear nuevos items en su recorrido mediante el análisis de brecha que se realiza al final de cada nivel.

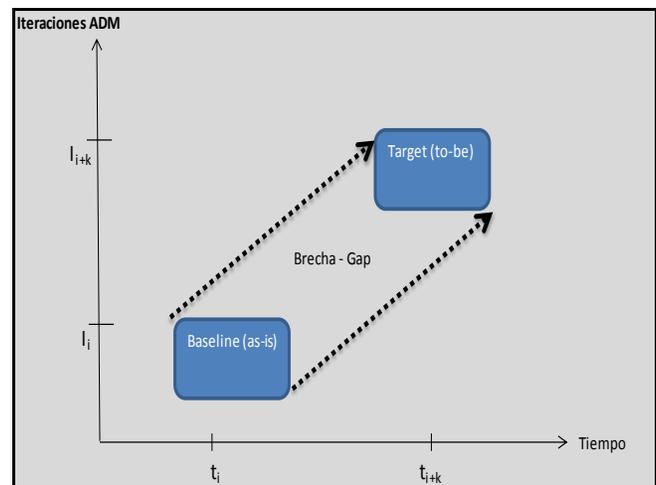


Figura 1. Iteraciones ADM y maduración.

El método permite transitar de una arquitectura inicial (baseline o as-is) hacia una arquitectura final objetivo (target o to-be) durante el proceso iterativo se realiza un análisis de brecha denominado GAP que mide

los objetivos de arquitectura y el grado de madurez alcanzados por la organización.[2]

El método ADM Figura2, consta de ocho (8) niveles o etapas de interacción y un paso preliminar que describe las actividades iniciales, principios y capacidades de la arquitectura objetivo, realiza además una personalización del marco de trabajo para ajustarlo a las necesidades de la organización.

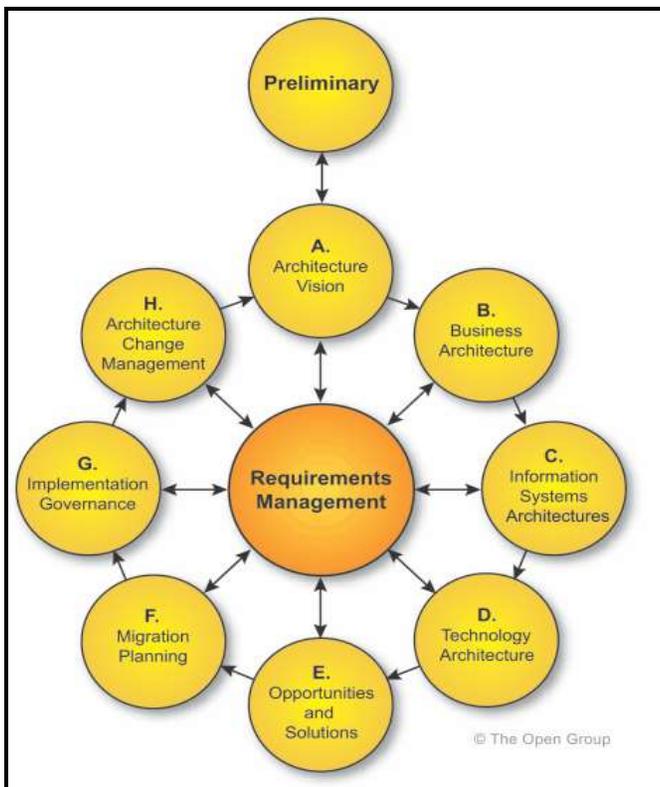


Figura 2. Método ADM.

Las fases B, C, D definen los cuatro niveles de abstracción para desarrollo de la arquitectura,

la fase E de la iteración define las oportunidades y soluciones que se deben implementar o migrar hacia un objetivo (target o to-be).

Estas oportunidades y soluciones son activos de la organización como procesos, tareas, normas, estándares, catálogos, procesos de negocio y por supuesto soluciones de software. Estos activos hacen parte del continuo empresarial que el marco de trabajo denomina Enterprise Continuum.

Los activos identificados y seleccionados son incluidos en el plan de migración (fase F) que tiene como finalidad su implementación y mover así la arquitectura de un estado base (baseline, as-is) a un estado objetivo (target, to-be).

Esta fase evidencia ya la inserción del método de desarrollo de software gracias al resultado de las fases previas que han especificado y entregado los requerimientos de un eventual producto de software.

Las fases G y H introducen gobernabilidad y gestión de control de cambios a la arquitectura. El ciclo se repite hasta lograr los

objetivos planteados en la visión de arquitectura (fase preliminar).[3]

MÉTODOS DE DESARROLLO SOFTWARE AGILES Y DISCIPLINADOS

Los métodos de desarrollo disciplinados enfocan su esfuerzo en lograr estabilidad, aseguramiento y predicción en el funcionamiento del software que se construye. Modelos como CMMI (Software Engineering Institute SEI), MSF (Microsoft), RUP (Rational Rose) se construyeron con un enfoque altamente estandarizado que define un lenguaje común de expresión e interpretación de los componentes del sistema, a nivel de patrones de la industria hacen parte de esta clasificación los estándares ISO/IEC 12207 (ISO, 2008) e IEEE 1074 (IEEE, 1998). RUP es uno de los estándares o marcos de trabajo más aceptados en la industria del software. Su visión es apalancar el desarrollo de proyectos de software en las organizaciones de acuerdo a sus necesidades y requerimientos.

RUP ha determinado un ciclo de vida del proyecto, que consta de cuatro fases. Estas fases deben permitir observar el proceso a un alto nivel de abstracción de manera similar a una metodología en cascada (waterfall).

El proceso principal se encuentra en las iteraciones de desarrollo de todas las fases del ciclo. Además, cada fase tiene un objetivo clave y un hito en el extremo que denota el objetivo que se logra. La visualización de las fases de RUP y sus disciplinas en el tiempo se expresan en el gráfico RUP Figura 3.

El método MSF de Microsoft comparte la visión de RUP en aspectos como la planeación y manejo del riesgo, implementados en artefactos como el documento de visión y las matrices de riesgo.

El artículo se detiene en el proceso RUP pero un análisis similar es posible sobre MSF, CMMi, IEEE o ISO. [4]

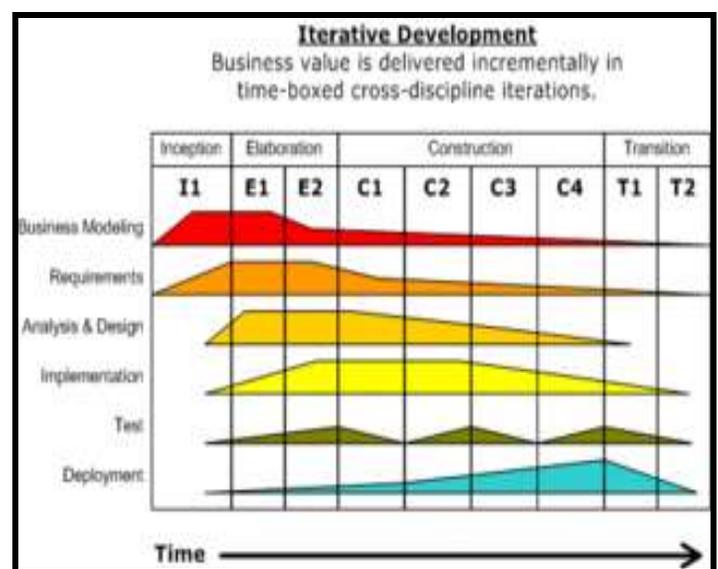


Figura 3. Proceso RUP fases y disciplinas.

El desarrollo ágil de software aplica métodos de ingeniería de software basado en procesos iterativos e incrementales, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos autos organizados y multidisciplinarios¹.

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen la forma de trabajo en equipos altamente productivos.

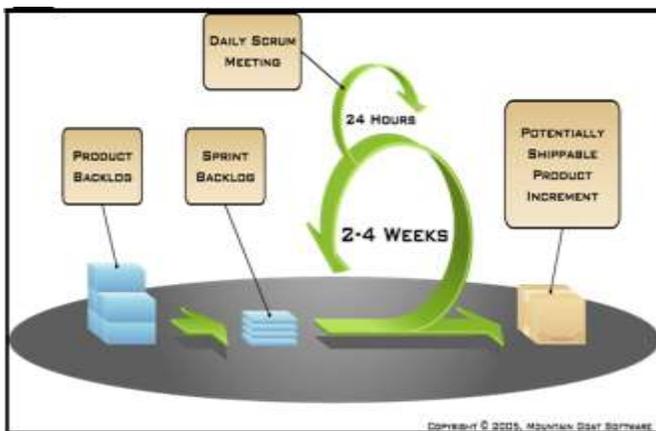


Figura 4. Proceso Scrum – Fuente: www.mountainsoftware.com

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

ENFOQUES DE INTEGRACION

TOGAF no provee guías o enlaces hacia algún proceso de desarrollo de software en particular (Rup, Xp, Scrum, Lean) el marco de trabajo TOGAF es agnóstico a las metodologías de desarrollo de software y el lineamiento que ofrece es integrar alguna de estas metodologías en las fases finales del método (F, G).

Enfoque de caja negra (black box approach): el proceso de desarrollo de software se enmarca dentro de las fases F (Migration Planing) y G (Implementation Governance) este enfoque toma ventaja del hecho que TOGAF no especifica un proceso de desarrollo e implementación[5]

¹ (Wikipedia, 2013)

Figura 5. Enfoque de caja negra ADM & MDS
Enfoque de caja blanca (white box): este enfoque es aplicable solo a procesos disciplinados de desarrollo como RUP, MSF o CMMi. En este enfoque la disciplina es incorporada al recorrer las fases del método

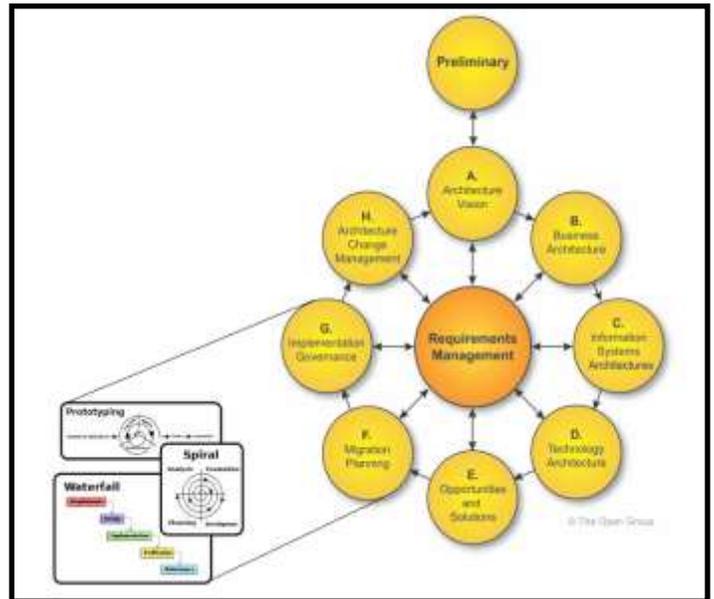
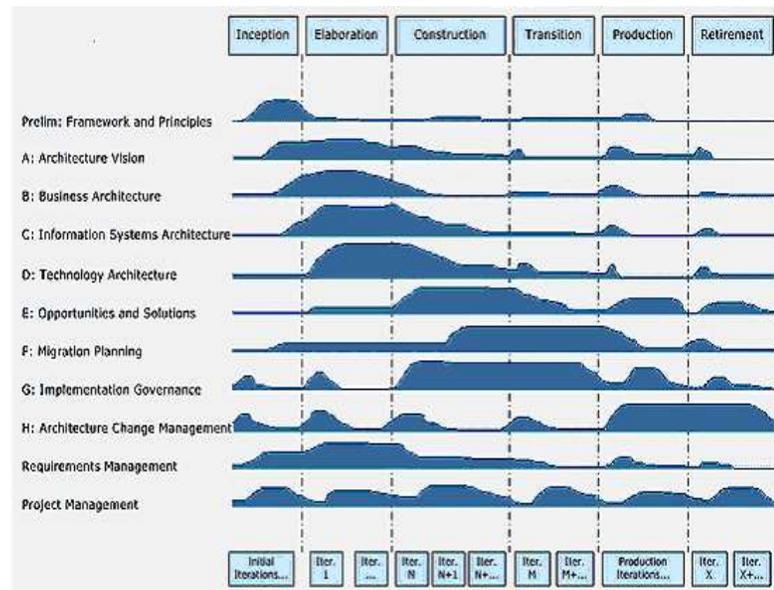


Figura 5. Enfoque de caja negra ADM & MDS

ADM de TOGAF Figura 6.

En la gráfica se observa que las fases del método de desarrollo definen las iteraciones sobre la disciplina ADM.



CONCLUSIONES

El marco de trabajo TOGAF ofrece gran flexibilidad en la selección del método de desarrollo de software. La decisión por parte del arquitecto empresarial y del arquitecto de software debe consultar variables como dimensión del producto a desarrollar, riesgos y estrategia de construcción, en particular si la solución es de gran alcance y tamaño se puede tomar la decisión de contratar a un tercero para que la decisión de la metodología y parte del riesgo en su implementación sea compartido, para soluciones pequeñas de entrega rápida y bajo riesgo se puede optar por desarrollos locales en la organización y el uso de metodologías ágiles como Scrum o Xp.

RECOMENDACIONES

La literatura actual sobre el tema es bastante incipiente y se evidencia una fisura entre la arquitectura empresarial y la arquitectura de soluciones de software, afortunadamente está surgiendo nuevas convergencias que incluyen híbridos metodológicos como DAD Disciplined Agile Delivery por sus siglas en inglés, que exploran y aprovechan lo mejor de los marcos y procesos actuales.

La invitación final es a continuar el estudio y exploración de un área importante para los profesionales de la Ingeniería de Software.

REFERENCIAS BIBLIOGRAFICAS

- [1] Naeem.naeemis.blogspot.com.
Recuperado el 2013, de
naeemis.blogspot.com.
<http://es.slideshare.net/carlos5paloma/togaf-7008349>
- [2] Samuli. (s.f.). www.samuliheljo.com.
Obtenido de www.samuliheljo.com:
<http://www.samuliheljo.com/blog/updated-myself-with-togaf-certification/>
- [3] Tarak Modi, V. &. (03 de 07 de 2008).
www.ebizq.net. Recuperado el 2013, de
www.ebizq.net:
http://www.ebizq.net/topics/soa_managem ent/features/9869.html?page=2
- [4] Asociación de arquitectos empresariales Iberoamérica: **(2013) Obtenido de**
<http://www.aogea.com.co/profiles/blog/list?tag=togaf>.
- [4] Process Corp., Framingham, MA. Intranets: Internet technologies deployed behind the firewall for corporate productivity. Presented at INET96 Annu. Meeting. [Online]. Available
- [5] Barry Boehm, R. T. (2003). Observations on Balancing Discipline and Agility. Observations on Balancing Discipline and Agility Discipline: A Guide to the Perplexed. Estados Unidos: Addison Wesley.