

XP/Architecture: Un Modelo Ágil para Escalar XP

Luis F. Muñoz, *‡ Julio A. Hurtado **

Fecha de recibido: 07/07/2012

Fecha de Aprobación: 25/11/2012

Resumen

XP es uno de los métodos ágiles más usados por la industria del *software* y la comunidad científica. Sin embargo, la literatura reporta problemas para escalarlo cuando el equipo crece y el producto es complejo. El SEI propone prácticas de arquitectura como mecanismos para resolver asuntos de complejidad y organización de las soluciones para alcanzar los atributos de calidad más relevantes. Aprovechando la capacidad de los métodos de arquitecturas y para separar preocupaciones a gran escala, en este artículo se propone un método de desarrollo de *software* en que varios equipos de desarrollo XP trabajan coordinadamente alrededor de un equipo que incorpora en forma ágil prácticas arquitectónicas. El método ha sido aplicado en un estudio de caso en dos proyectos académicos, mostrando la capacidad del modelo para escalar grupos trabajo, organizados en equipos XP en proyectos de mediana complejidad.

Palabras claves: *Ingeniería del Software, Procesos Software, Métodos ágiles, Arquitectura del software.*

Abstract

Reports about application of the Extreme Programming - XP method, have evidenced scalability problems in both, the team size and the problem complexity. When the requirements are more complex and the team size is bigger, the team organization, the project management and the product quality complexities are increased too, thus a typical agile methodology is not enough. Architecture practices deal with these concerns, so these are key elements for achieving any scalable XP extension. In this paper we propose XA (XP with Architecture), an XP extension including adapted architecture practices from architecture centered approaches. In order to evaluate the new model, the paper presents a case study where XA is applied in a two projects in an academic context. The results show as XA in the case study help to resolve scalability problems of XP teams working in a same software development project of medium-complexity.

* Fundación Universitaria de Popayán. programa de ingeniería de Sistemas. Estudiante de Doctorado en Ciencias de la Electrónica énfasis en Computación Universidad del Cauca. Popayán, Cauca, Colombia. lfreddy@fup.edu.co

** Universidad del Cauca. Departamento de Sistemas. Doctor en Ciencias de la computación Universidad de Chile. Popayán, Cauca Colombia. ahurtado@unicauca.edu.co

‡ Se concede autorización para copiar gratuitamente parte o toda el material publicado en la *Revista Colombiana de Computación* siempre y cuando las copias no sean usadas para fines comerciales, y que se especifique que la copia se realiza con el consentimiento de la *Revista Colombiana de Computación*

Keywords: *Software Engineering, Software Process, Agile Methods, Software architecture.*

1. Introducción

Gran parte de las compañías de *software* está compuesta por VSE (Very Small Enterprises)[1], donde la mayoría de las prácticas disponibles en la literatura no resultan aplicables directamente[2]. De ahí que, las metodologías ágiles estén dando una respuesta a la industria a través de desarrollos rápidos y de calidad en ambientes de gran incertidumbre. Las metodologías ágiles abordan de manera directa problemas de calidad que provienen de la comprensión de los requisitos [3], [4] a través de ciclos de desarrollo cortos, orientados al valor y con la participación activa del cliente. Extreme Programming – XP [4] como una de las metodologías ágiles más usadas en grupos pequeños, pero que presenta problemas de coordinación cuando el tamaño del equipo crece [5] evidenciándose el problema de la escalabilidad [6]. Escalar un método se refiere a extender su aplicación en un contexto más grande o complejo para el que fue propuesto inicialmente. Normalmente la escala está relacionada con la complejidad del problema y con el tamaño del equipo, por lo que cualquier estrategia de solución debe considerar elementos técnicos y de gestión con el fin de facilitar la descomposición metódica del proyecto en unidades más simples, gestionables y construibles. Aquí, la arquitectura es uno de los conceptos claves para resolver el problema en el que un proyecto de mediana o alta complejidad intenta usar metodologías ágiles [7]. La ausencia de una orientación hacia la arquitectura dentro de las metodologías ágiles, no permite que se tomen decisiones tempranas de diseño que tendrán un profundo impacto en todo el proyecto. Una industria de *software* competitiva debe poder incrementar su capacidad productiva a largo plazo, es decir, debe poder re-usar componentes, estructuras y decisiones de diseño, así como obtener productos que después puedan ser mantenidos adecuadamente [8]. Este artículo presenta *Extreme Programming with Architecture - XA*, un modelo de proceso de *software* basado en XP y centrado en la arquitectura, orientado a proyectos de mediana complejidad y equipos de más de 16 personas que busca potenciar la capacidad productiva de las organizaciones. XA introduce un modelo holístico que facilita la coordinación de un conjunto de equipos de desarrollo, pequeños y satélites que trabajan independientemente con XP alrededor de un grupo de arquitectura ágil. El modelo propuesto ha sido validado usando un caso de estudio embebido que involucra su aplicación en tres proyectos de desarrollo de *software*.

El resto de este artículo se ha organizado de la siguiente forma: la sección II presenta los principales trabajos relacionados a la escalabilidad de los

métodos ágiles, la sección III presenta el proceso XA, el cual involucra la especificación de los valores, equipos, prácticas y procesos. La sección IV presenta la aplicación del modelo en un caso de estudio embebido, así como el análisis de los resultados. Finalmente, la sección V presenta las conclusiones, las limitaciones y el trabajo futuro.

2. Trabajos Relacionados

Basado en experiencias reales, Kornstädt and Sauer [9] muestran que una vista arquitectónica común permite solucionar problemas de comunicación en un proyecto de desarrollo ágil. Aunque no se propone un proceso que combine métodos ágiles con métodos de arquitectura, se respalda la idea de que la arquitectura es un artefacto relevante para escalar un método ágil. Hadar y Silberman [9] presentan el método C3A, el cual está basado en el uso de diagramas de referencia y de implementación de la arquitectura en XP, un conjunto de contratos de componentes y una metodología que alinea el cronograma y la granularidad de las tareas. Sin embargo, este método no define explícitamente las prácticas respecto al desarrollo de los requisitos y el diseño de la arquitectura, tampoco reporta casos de aplicación. Jensen et al.[10] usan un mecanismo análogo a las historias de usuario, para representar los requerimientos y decisiones de arquitectura en XP con el objeto de plantear una arquitectura basada en las *historias de desarrolladores*. Estas historias expresan las demandas concretas de refactorización y son una herramienta para la planificación. Además, estas historias permiten a los desarrolladores, reflexionar sobre el diseño del sistema, para tener una comprensión compartida de la arquitectura. Sin embargo, las historias de desarrolladores no son usadas para definir una arquitectura de acuerdo con los requisitos de calidad, no se integran explícitamente prácticas de arquitectura, ni se presentan resultados empíricos. Donald J. Reifer et al.[11] recogen el criterio de varios expertos quienes dicen que no es posible escalar los métodos ágiles si no se usan metáforas mixtas (ágiles y rigurosas). Estos autores proponen un equipo de arquitectura y subdividir el resto del equipo en equipos más pequeños de desarrolladores, aclarando que dependiendo del tamaño del *software*, la comunicación y los otros principios del postulado ágil se pueden hacer tan complejos que en la relación cliente y equipos XP pueden perderse los objetivos del sistema.

Respecto a los estudios en los que se mide la productividad y a calidad de los desarrollos con XP, se encuentra muy poco reportado en la literatura. [12] reportan un estudio de caso en el que se alcanzó una productividad de 0,003 historias de usuario persona hora (32 historias de usuario, en 3.5 meses por un equipo de 14.7 personas). Otro estudio de caso de estudio es reportado por [13], donde la productividad de un equipo XP es de 17 NLOC-P-(NLO-P-H - Número de líneas de código por persona por hora)

3. Metodología

3.1. Extensión de XP con Prácticas de Arquitectura

XA es una metodología ágil sostenida por los valores y principios propios de la programación extrema (XP), que agrega algunas prácticas de arquitectura, cuyo objeto primordial de hacer usable XP con proyectos de mediana complejidad (problema/solución) y con equipos más grandes a los tradicionales [21]. La figura 1 muestra como el gran equipo del proyecto es descompuesto por un conjunto de equipos XP más pequeños alrededor de un equipo de arquitectura XA. Así, las dinámicas del modelo XA son una combinación de las dinámicas presentes en estos grupos y sus interrelaciones. Por ello la descripción de este modelo holístico parte de las prácticas y valores de XP y estas trascienden a la dinámica intra-grupo del grupo de arquitectura y a la dinámica inter-grupo del equipo completo.

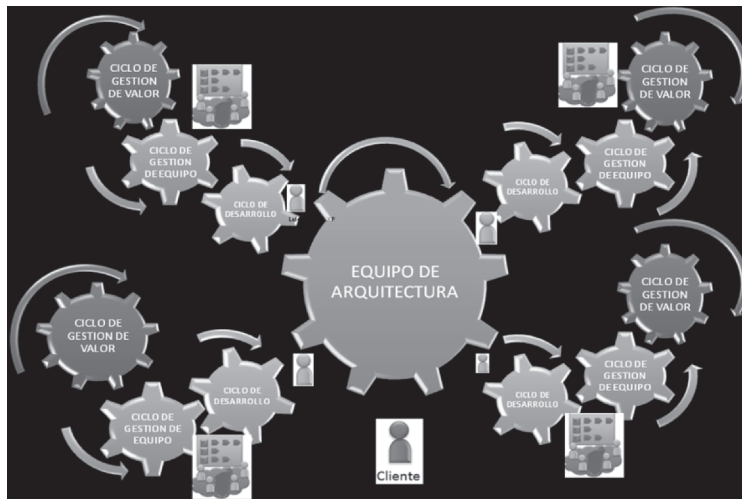


Fig. 1. Modelo Holístico de XA.

Por lo anterior, para XA son muy importantes los 4 valores propuestos por el manifiesto ágil [4], y además XA propone un nuevo valor que formará parte de los pilares que orientaran un desarrollo ágil en proyectos de mediana complejidad:

- La organización de equipos XP: Debido a la magnitud del proyecto, este nuevo valor se orienta a la conciencia sobre organización holística de equipos XP, donde cada equipo sigue los valores y principios de XP [7] y al menos un equipo tendrá más énfasis en la arquitectura de software, el cual seguirá las prácticas extendidas

definidas en XA. A los equipos que siguen los principios de XP se denominan equipos XP y al equipo que sigue además las prácticas de arquitectura se denomina equipo XA.

- La comunicación: Basada en una alta interacción entre los equipos de trabajo XP, los miembros del equipo XA y de estos con el cliente, la comunicación requiere establecer los canales para que el conocimiento fluya de forma emergente. El cliente debe ser miembro activo del equipo XA como fuente permanente de los requisitos del sistema. Este equipo está conformado por un miembro de cada equipo XP, quienes a su vez hacen de clientes en su respectivo equipo XP. Al cliente del proyecto se le denominará Cliente XA y al cliente de cada equipo XP, miembros a su vez del equipo XA, se le denominará Cliente XP. Los clientes XP son agentes que establecen la relación inter-grupo y por tanto son los elementos claves de la comunicación en XA.
- La simplicidad: Aplicada a todos los aspectos de XA, desde la fase inicial con arquitecturas muy sencillas y basadas en prácticas básicas, incluyendo la refactorización de código. Para XA, lo importante es hacer lo que el cliente necesita y de la forma más sencilla, pero sin perder la posibilidad de la integración continua y de valor.
- La retroalimentación: Sucede en dos momentos; el primero por parte del equipo XA hacia el cliente XA, con el fin de brindarle avance sobre el desarrollo del sistema y conocer su apreciación, y desde el cliente XP hacia el equipo XP en los aportes al desarrollo del proyecto. El segundo momento ocurre entre cada miembro de los equipos XP, y los miembros del equipo de arquitectura XA, con el fin de brindarle avance sobre el desarrollo del sub-sistema y conocer su apreciación, y desde el cliente XP hacia el equipo XP en los aportes al desarrollo del sub-proyecto.
- La valentía: Para enfrentarse a los continuos cambios que se presentan en el transcurso de la actividad, que implica a todos los miembros tanto del equipo XA como los equipos XP.

Además de las doce (12) prácticas que caracterizan a XP [14], existen otras que caracterizan XA:

- Gestión ágil con arquitectura: cada equipo XA y XP tienen un líder de proyecto quienes lo gestionan de forma ágil. Para la integración de las prácticas de gestión el líder del proyecto XA se reúne periódicamente (reunión diaria de gestión) con los líderes de cada proyecto XP para abordar los problemas y enfoques de gestión del proyecto.

- Unificación de código en equipos XA: Al igual que con los equipos XP, existe para el equipo de arquitectura una propiedad colectiva de código [15], pero con el objeto de hacer menos complejo el desarrollo y cumplir con los criterios ágiles en cada equipo existe la propiedad colectiva del código del sub-sistema que tiene a cargo (la propiedad colectiva de código se mantiene como práctica en cada equipo). XA se apoya de prácticas de arquitectura para la distribución de subsistemas, sus requisitos y posterior validación e integración. XA puede verse como un equipo ágil que subcontrata el desarrollo de componentes con equipos XP.
- Mecanismo para distribuir requisitos: Una vez establecido un incremento en la arquitectura, por el equipo XA, se le comparte a los equipos XP los requerimientos del subsistema aprovechando la interacción de un miembro XA como cliente del equipo XP y por tanto manteniendo la filosofía de XP.

3.2. El Ciclo de Vida de XA

XA sigue unas fases e iteraciones como se muestra en la Figura 2



Fig. 2. Fases del método XA.

- Exploración: se reúnen los miembros del equipo XA con el cliente (XA) para establecer los requisitos iniciales del sistema. En esta fase se delimita el sistema.
- Planificación: los miembros del equipo XA establecen los objetivos de la fase, y construyen versiones iniciales de la arquitectura del sistema.
- Construcción con arquitectura: Basados en la fase anterior se priorizan los requisitos del sistema y se desarrolla un esqueleto de la arquitectura planteada por el equipo de arquitectura para esos requisitos, se modulariza el sistema y se distribuyen los requisitos a los equipos XP que participen en el proyecto. La definición de arquitectura se desarrolla paralelo al desarrollo de los equipos XP. Continuamente se va integrando el sistema. Como en XP esta construcción se realiza de manera iterativa e incremental garantizando la retroalimentación continua del cliente.

- Entrega Final: El equipo XA hace entrega al cliente del producto final, aunque el ha realizado validaciones incrementales; debe evaluar completamente el producto.

Una iteración de XA se muestra en la Figura 3. En esta iteración de XA se ha embebido la iteración de XP, la cual es ejecutada por el equipo XP. Las actividades principales se han paralelizado con el objetivo de dar libertad a los equipos de ejecutar estas tareas de acuerdo con la dinámica emergente.

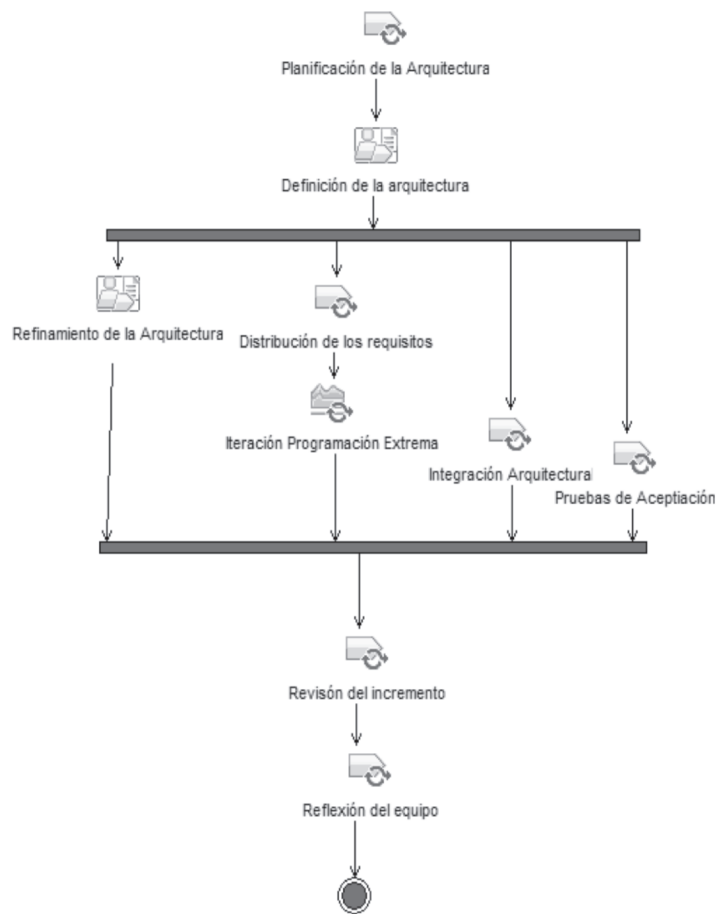


Fig. 3. Una Iteración en el Método XA.

3.3. Prácticas de Arquitectura en XA

XA mantiene de XP la simplicidad, la metáfora del sistema y la refactorización. Además se aplican en forma adaptada los métodos de

arquitectura: *Attribute-Driven Design ADD* y [16] QualityattributeWorkshops (QAW) [17]. Estas prácticas han sido combinadas y compactadas con las prácticas base de XP en trabajos previos [8],[18] y esto ha sido tomado como punto de referencia para obtener a través de esta investigación los fragmentos de proceso referentes a temas arquitectónicos como se muestra a continuación:

- *Juego de la Planificación*: Las historias de usuario que resultan de la dinámica de XP, se complementan con información de atributos de calidad proporcionados por el QAW donde se tienen en cuenta los escenarios de calidad, denominados en *XA historias de arquitectura*. La priorización de las historias de arquitectura y el refinamiento de las mismas dan información adicional para que el cliente y los desarrolladores elijan las historias de usuario en cada iteración. En la fase de exploración y de planificación se identifican y seleccionan las historias de arquitectura claves para arquitectura, muy útiles en la elección de las historias de usuario durante la práctica del juego de planificación de los equipos XP.
- *Cliente in situ*: El concepto de cliente es extendido para dar cabida a la participación de otros participantes (ej. equipo de mantenimiento, gerencia) con el fin de verificar los requisitos del sistema y las soluciones técnicas más allá de un dueño de los requisitos de negocio. Esto se hace en algunas de las reuniones diarias del equipo XA. Cada equipo XP mantiene esta práctica, en que su cliente es el equipo de arquitectura.
- *Metáfora del sistema*: El método ADD proporciona un enfoque de refinamiento recursivo para la definición de la arquitectura en términos de descomposición del sistema en módulos más pequeños. XA usa este enfoque para descomponer el sistema en forma *simple y alegórica al mundo del problema* y brindar así un mapa comprensible del producto para distribuir los requisitos.
- *Diseño Simple*: El método ADD proporciona una arquitectura en un alto nivel de abstracción suficiente para garantizar un diseño competente, y las historias de usuario y de arquitectura, de tal forma que se mitiguen los riesgos asociados al proyecto. Contar con un diseño de alto nivel facilitará la integración y la independencia de los equipos XP en sus diseños, para que estos puedan mantener esta práctica intacta en su interior.
- *Refactorización*: En lo arquitectónico facilita la formulación rápida de soluciones mientras que en paralelo ADD brinda tácticas y estrategias de arquitectura para la solución a más largo plazo, disparando

requisitos de refactorización arquitectónica. Esto facilita que los requerimientos técnicos puedan surgir incrementalmente respaldado por la capacidad de las prácticas de negociación de requisitos y refactorización en los equipos XP. Las refactorizaciones en XA, son de tipo arquitectónico con el fin de ubicar los cambios potenciales de mayor impacto lo más temprano posible en el proyecto.

- *Propiedad colectiva de código*: Mantiene cada equipo XP a través de su subsistema en desarrollo. El equipo XA, define en mayor medida elementos de código arquitectónicos y sincroniza las tareas realizadas por los equipos XP; para ello es necesario tener un repositorio de aplicaciones y de manejo de versiones que garantice el trabajo sobre estas restricciones.
- *Desarrollo dirigido por pruebas*: Es extendido por el desarrollo de pruebas tempranas de integración. Las pruebas tempranas de integración son definidas para probar el comportamiento abstracto de la arquitectura antes de que este se implemente, brindando una idea de los mecanismos que deberán ser definidas como una forma de requisitos de implementación. Las pruebas son ejecutadas implementando el comportamiento abstracto en forma concreta con código ejemplo (requerido por el caso de prueba) antes de que este comportamiento sea desarrollado por el equipo XA y liberado a los equipos XP.

3.4. Actividades Principales de XA Respecto a la Arquitectura

Al igual que XP[19], XA propone unas actividades que forman parte de su ciclo de vida y que ponen en evidencia las prácticas anteriormente descritas en las siguientes actividades que se desarrollan en forma concurrente:

- *Analizar requisitos arquitectónicos*: esta actividad basada en criterios de calidad de QAW en la que se identifican y analizan las historias de la arquitectura con el fin de abordar lo más tempranamente posible temas de priorización y de descomposición del sistema. Para identificar historias del arquitecto, el equipo XA identifica y define las historias de usuario y de arquitectura, basadas en esquemas y artefactos no formales, visibles a través del radiador de información del equipo XA. Durante el juego de la planificación, las historias de arquitectura son priorizadas y expresadas mediante un objetivo medible. Las historias de usuario y los requisitos de calidad en las historias de arquitectura, las cuales especifican una situación directa como el desempeño, la seguridad y la disponibilidad, así como situaciones indirectas como la modificabilidad y la escalabilidad.

- **Diseñar la arquitectura:** El diseño de la arquitectura parte de planteamientos iniciales llamados Arquitecturas candidatas. En estas, el equipo XA plantea esbozos iniciales de arquitecturas basado en el conocimiento del negocio, de la tecnología y de sus experiencias previas. Las arquitecturas candidatas son evaluadas utilizando los escenarios de las historias de arquitectura con el objetivo de determinar su viabilidad y factibilidad. De acuerdo con la evaluación, se hace una selección de las arquitecturas candidatas, donde el equipo de arquitectura tomará como arquitecturas candidatas aquellas que más favorezcan el desarrollo del proyecto. Basadas en estas arquitecturas candidatas, la arquitectura del sistema es detallada exhibiendo un conjunto de decisiones de diseño que serán puestas en el radiador de información, además de un conjunto pequeño de vistas simples y comprensibles a los distintos involucrados.
- **Implementar la arquitectura:** consiste en implementar las interfaces, el comportamiento abstracto y documentar las API con el fin de convertirlos en los principales artefactos de descripción de la arquitectura. La implementación sigue la estrategia del desarrollo dirigido por pruebas: una vez desarrollado el concepto de arquitectura, cada equipo XP codifica las pruebas de sus diseños y sus diseños mismos, de tal forma que desde muy temprano se pueda mostrar al cliente módulos funcionales. Las interfaces, además de definir en la programación la firma de los servicios, deberá asociarle información referente a la calidad de los mismos, sus restricciones y las decisiones que condujeron a dicha especificación. La implementación de la arquitectura requiere además implementar y documentar en el código las pruebas, el comportamiento abstracto y el comportamiento concreto esquelético. La documentación de la API es necesaria para que el equipo XA presente a los equipos XP, las estrategias y especificaciones surgidas de las decisiones arquitectónicas. La documentación del mismo código se fundamenta en los criterios ágiles de XP, y se realizará de forma incremental a medida que el código arquitectónico y no arquitectónico crece.

4. Estudio de Caso

XA fue aplicado en estudio de caso embebido en el marco de dos proyectos académicos, manteniendo las directrices de la investigación empírica de los estudios de caso donde se estudia un fenómeno dentro de su contexto real [9].

4.1. Diseño del Estudio de Caso

El objetivo de este estudio de caso de estudio fue evaluar la capacidad de XA para escalar a grupos grandes en el marco de un proyecto de desarrollo grande respecto a un tiempo corto de desarrollo. Para diseñar esta investigación, se siguió la propuesta de Runenson y Host [20], para ello se partió de la pregunta de investigación principal que se intenta resolver: ¿Cómo escalar un proceso ágil en proyectos de mediana complejidad y equipos de desarrollo medianos? junto con la revisión de la literatura científica relacionada, se definió como hipótesis: *XA mantendrá en el caso de estudio el orden de los índices de productividad y satisfacción reportados para XP, pero en proyectos de mediana complejidad y con equipos de más de 10 ± 2 personas*. Para soportar esta investigación se definieron los indicadores, métricas e instrumentos de recolección de datos como se presentan en la Tabla 1.

INDICADOR	MEDIDAS	FUENTES DE INFORMACIÓN	INSTRUMENTOS
Satisfacción	NA- Nivel de Aceptación NP- Nivel de Participación	Cliente y Equipo	Encuesta
Calidad	NC- Nivel de Conformidad NS- Nivel de Satisfacción	Cliente y Producto	Encuesta y Registro de defectos
Productividad	P- Productividad	Artefactos de Gestión	Gráfico Burndown

Tabla 1. Indicadores, Métricas, Fuentes de información, Instrumentos.

Para la recolección de la información en este estudio de caso se usaron y diseñaron los siguientes instrumentos:

- Encuestas: aplicadas al cliente, equipos XP y equipo XA, con el fin de indagar el grado de participación del cliente, su compromiso respecto al proyecto y con equipo XA, así como su satisfacción con el método y el producto. Conocer el grado de aceptación del modelo propuesto entre los miembros del equipo y el grado de receptividad del modelo y su dinamismo frente al proyecto y a los otros actores.
- Artefactos: como la lista de chequeo del producto para evaluar el cubrimiento de los requisitos, la planilla de defectos para evaluar la calidad y la evaluación de la arquitectura para evaluar la calidad de la arquitectura propuesta.

4.2. El Caso de Estudio

XA en un primer momento se aplicó a un proyecto denominado Gestor de Proyectos de Investigación del Programa de Ingeniería de Sistemas de la Fundación Universitaria de Popayán [21], con estos resultados se

decidió aplicar el modelo en otro proyecto para evidenciar otras variables que debería ser controladas.

4.2.1. Descripción del Proyecto

Para este caso de estudio, se aplicó XA a un grupo de 18 estudiantes quienes se caracterizaron porque el 60% de ellos habían culminado materias y realizaban con la Institución un diplomado en desarrollo de software y el 40% restante egresados de programas de Ingeniería de Sistemas de diferentes Universidades de la ciudad interesados en el tema y en la propuesta de trabajo para el desarrollo del proyecto. A este grupo, se les propone también el desarrollo de una aplicación de gran escala. El proyecto a desarrollar igual que con el primer grupo consistió en una aplicación distribuida para la inscripción y seguimiento de proyectos de investigación. El desarrollo lo realizaron con tecnologías PHP y la librería JQUERY-UI, para la base de datos MySQL y como servidor web Apache. Nuevamente para este caso, el cliente fue un profesor designado por el comité curricular, representante del programa de Ingeniería de Sistemas ante el Centro de estudios e Investigaciones y quien conocía la dinámica del proyecto y los requisitos para el desarrollo de la aplicación.



Fig. 4. Reunión XA.

Desarrollo del proyecto: Para la aplicación de XA, (Figura 4) los estudiantes recibieron una capacitación inicial de 12 horas dividida en tres partes. La primera parte abordó la temática de las metodologías

ágiles (XP), con el objeto de unificar términos de referencia. En la segunda parte se abordó el tema de arquitectura, particularmente en lo que respecta sus conceptos, vistas, estilos y a los métodos ADD y QAW. En la tercera parte, se les presentó XA, dando a conocer al grupo los parámetros e instrumentos que se llevarían a cabo para el proyecto del curso. Para respaldar la capacitación y el desarrollo del proyecto se les entregó dos reportes técnicos desarrollados específicamente para introducir XP y XA. El proyecto se desarrolló en clases con sesiones de 4 horas cubriendo un total de 48 horas.

Los estudiantes se auto-organizaron en 3 equipos XP de 6 integrantes, cada equipo escogió su representante cliente/arquitecto ante XA. El equipo XA fue organizado por los tres representantes de los equipos XP. En la dinámica del proyecto, mientras XA se reunió con el cliente del proyecto, los equipos XP organizaron sus radiadores de información (carteleros) en un sitio visible para todos. En todas las sesiones (12) de 4 horas cada una, se desarrolló al comienzo una reunión de 15 minutos entre el cliente y el equipo XA para recibir retroalimentación del cliente. El cliente tuvo un compromiso muy alto, estuvo disponible dentro de la institución y participó activamente en el desarrollo de los requisitos. Además los equipos XA y XP siguieron las prácticas de planificación de iteración y del trabajo diario de acuerdo a lo establecido por XP y XA. Durante el desarrollo del proyecto se recolectó la información de acuerdo a lo planificado. La dinámica corresponde al modelo planteado, puesto que se notó durante el desarrollo del caso un compromiso por cada uno de los miembros del equipo al seguir prácticas propuestas por el modelo XA, lográndose una adherencia del 82%, el cual puede considerarse aceptable para poder evaluar los resultados del caso de estudio.

5. Resultados

La productividad de los equipos XP fueron entre 0.010 y 0.017 HU-P-H, mientras los equipos XA fueron de 0,007 HA-P-H respectivamente, lo cual es notablemente mejor a lo reportado por [12] donde la productividad fue de 0,003 HU-P-H (32 historias de usuario, en 3.5 meses por un equipo de 14.7 personas). Por otro lado de acuerdo a [13], la productividad de un equipo XP es de 17 NLOC-P-H (Número de líneas de código por persona por hora) fue notablemente mejor que un proyecto desarrollado en forma tradicional (10.3 NLOC-persona-hora). En este caso de estudio, se desarrollaron 12600 NLOC alcanzando una productividad de 14.58 NLOC, menor a la reportada por [13] aplicando XP, pero manteniendo el orden de productividad y mejor que la reportada usando el enfoque tradicional. Estos resultados permiten validar preliminarmente la hipótesis de que la productividad de los

equipos XA mantendría el orden de productividad de los equipos XP reportados por la literatura. Sin embargo, una mayor evaluación empírica debe ser desarrollada.

INDICADOR	PROYECTO 1	PROYECTO 2
Nivel de Aceptación de XA	100%	100%
Nivel de Participación del Cliente	95%	97%
Nivel de conformidad del producto	70%	87%
Satisfacción de los requisitos	95%	97%
Productividad de los equipos XP	0.010 HU-P-H	0.017 HU-P-H
Productividad de los equipos XA	0.007 HA-P-H	0.018 HA-P-H
Adherencia al método XA	82%	93%

Tabla 2. Resultados.

6. Conclusiones, Limitaciones y Trabajo Futuro

En este artículo se ha presentado XA, un método de desarrollo que extiende las prácticas de XP y las escala a equipos más grandes y proyectos más complejos. De forma preliminar el método escala al aplicarlo en un caso académico, al compararlo con otros estudios. Por tanto, es necesario verificar el modelo en un contexto industrial, donde las variables que se presentan en este tipo de ejercicios son bastante exigentes para ser controladas. Dentro de los principales hallazgos cualitativos de este estudio de caso se tiene:

- La comunicación entre el equipo XA y los equipos XP es una de las partes de mayor riesgo en un proyecto con XA, por lo tanto se debe priorizar este aspecto buscando que esta sea fluida, clara y coherente. Los miembros del equipo XA deben conocer y manejar claramente los objetivos del proyecto y los requisitos de negocio del cliente.
- Los representantes del equipo XA deben no solamente ser el elemento de conexión entre el equipo de arquitectura y los equipos XP, sino que deben ser los dinamizadores y motivadores del equipo.
- Al final de cada iteración, hubo un nuevo conocimiento (valor agregado) aprendido por cada miembro de los equipos. Los miembros del equipo XA comprendieron su papel para lograr la integridad del modelo holístico, en generar las historias de arquitecto como mecanismos para el establecimiento de los requisitos de todos los equipos XP.

La investigación tiene algunas limitaciones debido al tipo de comparaciones de las mediciones realizadas y a las encontradas en la literatura, dado que la literatura en la mayoría de casos no las reporta o

las reporta sin caracterizar el contexto de los estudios de caso. Para ello se busca en una siguiente fase de este trabajo realizar un experimento controlado para poder establecer una comparación más aproximada y más confiable. Además, se espera aplicar XA en casos de la industria, donde se obtendría una retroalimentación más enriquecida. El caso de estudio aquí presentado ha permitido establecer que para corroborar los resultados obtenidos en esta investigación será necesario considerar otras variables de contexto como las establecidas por [12] para hacer una comparación más objetiva, por ejemplo, la experticia de los participantes. Todas estas consideraciones están actualmente siendo adoptadas para mejorar el diseño de futuros estudios de caso y el caso experimental.

Referencias

- [1] Fedesoft. (2011) Noticias TIC. [Online]. Available: <http://www.fedesoft.org>
- [2] P. L. José Hilario Canós and C. Penadés, “Metodologías Agiles en el Desarrollo de Software,” Tech. Rep., 2003.
- [3] R. J. Costello and D.-B. Liu, Metrics for Requirements Engineering. *Journal of Systems and Software* 1995, 1995.
- [4] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change* (2nd Edition). Addison-Wesley Professional, 2004.
- [5] P. C. Pendharkar, Rodger, and J. A., “The Relationship Between Software Development Team Size and Software Development Cost,” *Commun. ACM*, vol. 52, no. 1, pp. 141–144, Jan. 2009.
- [6] R. L. Nord, Tomayko, and J. E., “Software Architecture-Centric Methods and Agile Development,” *IEEE Software*, vol. 23, no. 2, pp. 47–53, Mar. 2006.
- [7] E. Hadar and G. M. Silberman, “Agile architecture Methodology: Long Term Strategy Interleaved with Short Term Practics,” in *Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications*, ser. OOPSLA Companion '08. New York, NY, USA: ACM, 2008, pp. 641–652.
- [8] O. S. P Abrahamsson and J. Ronkainen, *Agile Software Development Methods: Review and Analysis*. VTT Electronics, 2002.

- [9] A. Kornstadt and J. Sauer, "Tackling Offshore Communication Challenges with Agile Architecture-Centric Development," in Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture, ser. WICSA '07. Washington, DC, USA: IEEE Computer Society, 2007, pp.
- [10] P. S. Rolf Njor Jensen, Thomas Maller and G. Tarnehaj, Architecture and Design in eXtreme Programming; Introducing Developer Stories. Lecture Notes in Computer Science, 2006.
- [11] D. J. Reifer, F. Maurer, and M. H. Erdogmus, "Scaling Agile Methods," IEEE Software, vol. 20, no. 4, pp. 12–14, 2003.
- [12] L. Layman, L. Williams, and L. Cunningham, "Exploring Extreme Programming in Context: An Industrial Case Study," in Proceedings of the Agile Development Conference, ser. ADC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 32–41.
- [13] F. Maurer and S. Martel, "On the Productivity of Agile Software Practices: An Industrial Case study," International Workshop on Economics-Driven Software Engineering Research (EDSER, Tech. Rep., 2002.
- [14] D. Wells, "Extreme Programming a Gentle Introduction," 2012. [Online]. Available: <http://www.extremeprogramming.org>
- [15] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, and W. Cunningham, Manifesto for Agile Software Development. URL: <http://agilemanifesto.org/>, 2009.
- [16] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord, and B. Wood, Attribute-Driven Design (ADD), Version 2.0. Software Engineering Institute, 2006, no. CMU/SEI-2006-TR-023.
- [17] M. R. Barbacci, R. J. Ellison, A. J. Lattanze, J. A. Stafford, C. B. Weinstock, and W. G. Wood, "Quality Attribute Workshops QAW -Third Edition," Carnegie Mellon, Technical Report CMU/SEI-2003-TR-016, Oct. 2003.
- [18] M. Freddy and H. Julio, "XP/Architecture," Tech. Rep. IDIS-TR-002, 2011.
- [19] T. Dybå and T. Dingsøy, "Empirical Studies of Agile Software Development: A Systematic Review," Inf. Softw. Technol., vol. 50, no. 9-10, pp. 833–859, Aug. 2008.

- [20] P. Runeson and M. Höst, “Guidelines for Conducting and Reporting Case Study Research in Software engineering,” *Empirical Softw. Engg.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.
- [21] Muñoz Sanabria Luis Freddy. Hurtado Julio Ariel. XA: An XP extension for supporting architecture practices Computing Congress (CCC), 2012 7th Colombian Digital Object Identifier: 10.1109/ColombianCC.2012.6398012. 2012.