

Artículo original producto de la investigación.

Guía metodológica ágil, para el desarrollo de aplicaciones móviles “AEGIS-MD”

Yohn Daniel Amaya Balaguera ¹

Resumen

Objetivo. Desarrollar una guía metodológica basada en características de las metodologías XP, SCRUM y TDD con la finalidad de proporcionar elementos conceptuales, pasos estructurados, roles y herramientas a ser aplicadas en el desarrollo de aplicaciones para dispositivos móviles, con el fin de aumentar la productividad en el proyecto y asegurar la calidad del producto entregado.

Método. La investigación se dividió en tres fases; en la primera, se realizó una revisión de las metodologías ágiles para desarrollo de software y la identificación de los requisitos específicos de las aplicaciones para dispositivos móviles; en la segunda fase, se realizó una caracterización del desarrollo de aplicaciones para dispositivos móviles en la etapa de análisis, diseño e implementación, así como una revisión de las herramientas de gestión de calidad que ofrecen las metodologías ágiles y son susceptibles a ser implementadas en el desarrollo de aplicaciones para dispositivos móviles, en la tercera fase; se identificaron algunas prácticas de las metodologías ágiles, susceptibles a ser aplicadas en el desarrollo de software para aplicaciones móviles.

Resultados. Estructuración de una guía metodológica.

Palabras clave: aplicación móvil, desarrollo software, metodología ágil.

1. Magister en Tecnología Informática, Universidad Pedagógica y Tecnológica de Colombia.
Correo: yohn.amaya@uptc.edu.co
ORCID: <http://orcid.org/0000-0003-1894-7887>

Agile methodological guide for the development of mobile applications "AEGIS-MD"

Abstract

Objective. Develop a methodological guide based on characteristics of XP, SCRUM and TDD methodologies in order to provide conceptual elements, structured steps, roles and tools to be applied in the development of applications for mobile devices, in order to increase productivity project and ensure the quality of the product delivered.

Method. The research was divided into three phases; in the first, a review of agile software development methodologies and identification of the specific requirements of mobile applications was made; in the second phase, a characterization of the development of applications for mobile devices in the stage of analysis, design and implementation, as well as a review of the quality management tools that offer agile methodologies was made and are likely to be implemented in the application development for mobile devices, in the third stage; some agile practices, likely to be applied in the development of software for mobile applications identified methodologies.

Results. Structuring a methodological guide.

Keywords: Agile methodology, mobile application, software development.

Recibido: 23-04-2015

Aceptado: 12-05-2015

Introducción

Durante la última década el auge del software se ha centrado principalmente en el desarrollo e implementación de aplicaciones para dispositivos móviles; sin embargo, pese a su gran fulgor y a la existencia de muchas metodologías de desarrollo ágiles, las cuales, gozan de buena acogida actualmente entre los desarrolladores (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003), («State of Agile Development Survey Results | VersionOne», s. f.), casi ninguna se ha centrado en los requerimientos tan específicos que el desarrollo móvil necesita,

ya que, la creación de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos (Satyanarayanan, 1996), lo que provoca que las metodologías usadas para estos entornos también difieran de las del software tradicional; esto es, porque el software para dispositivos móviles tiene que satisfacer una serie de requerimientos y condicionantes especiales tales como: canal, movilidad, portabilidad, capacidades específicas de las terminales, entre otras (Pekka Abrahamsson, 2005), (Pekka Abrahamsson *et al.*, 2004), las cuales hacen de su desarrollo un proceso más complejo, el cual, en la mayoría de los casos es tomado por esfuerzos individuales de desarrollo, los cuales hacen más difícil su integración con la ingeniería de software y es por eso que la adaptación de las metodologías ágiles en el desarrollo de aplicaciones para móviles es una importante inversión a la hora de contribuir al mantenimiento de la calidad en las aplicaciones para dispositivos móviles, aunque sus características específicas demandan importantes ajustes sobre las actuales metodologías ágiles.

Con base en lo anterior, esta propuesta presenta una adaptación híbrida de las mejores herramientas y procesos que presentan las metodologías ágiles XP, SCRUM y TDD, pero enfocadas en los requerimientos específicos de las aplicaciones móviles. En la primera parte de este documento se exponen algunas características de las metodologías ágiles usadas en el desarrollo de la guía metodológica, además de una breve revisión de las metodologías utilizadas actualmente en el desarrollo de aplicaciones para dispositivos móviles; en la segunda parte se expondrán algunas generalidades del desarrollo de aplicaciones para dispositivos móviles, para concluir con la presentación de la guía metodológica ágil, para el desarrollo de aplicaciones para dispositivos móviles AEGIS-MD.

Fundamento teórico

A continuación se presenta una breve descripción sobre las metodologías ágiles XP, SCRUM y TDD, tomadas como base para el desarrollo de la investigación, ya que, cuentan con buena aceptación dentro de los equipos de desarrollo (Salo & Abrahamsson, 2008), (Abrahamsson *et al.*, 2003), («State of Agile Development Survey Results | VersionOne», s. f.) y son adaptables a las características específicas del desarrollo de aplicaciones para dispositivos móviles.

Metodologías ágiles

Las metodologías en general se clasifican según su enfoque y características esenciales, las más recientes, que se fueron gestando a finales del siglo pasado y que se han comenzado a manifestar desde principios del actual, se han denominado “metodologías ágiles” y surgen como una alternativa a las tradicionales; estas metodologías se derivan de la lista de los principios que

se encuentran en el "Manifiesto Ágil" («Manifiesto por el Desarrollo Ágil de Software», s. f.), que están basados en un desarrollo iterativo que se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones de diferente longitud, cada una de ellas generando un producto completo y entregable; donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto, las interacciones individuales deben producir alguna característica completamente funcional o mejorada (Szalvay, 2004) su principal objetivo es reducir el tiempo de desarrollo.

La aparición de las metodologías ágiles no puede ser asociada a una única causa, sino a todo un conjunto de ellas; si bien es cierto que la mayoría de autores lo relacionan con una reacción a las metodologías tradicionales, ¿Cuáles fueron las causas de esta reacción?, los factores que comúnmente se mencionan son la pesadez, poca o nula aceptación al cambio, lentitud de reacción y exceso de documentación, en definitiva, falta de agilidad de los modelos de desarrollo formales.

Extreme Programing (XP)

Se centra en las mejores prácticas para el desarrollo de software. Consta de doce prácticas: el juego de planificación, pequeñas emisiones, la metáfora, el diseño sencillo, las pruebas, la refactorización, la programación en parejas, la propiedad colectiva, integración continua, semana 40-h, los clientes en el lugar, y los estándares de codificación (Beck & Zapata Martínez, 2002). La versión revisada "XP2" se compone de las siguientes prácticas "primarias": sentarse juntos, equipo, espacio de trabajo informativo, el trabajo de energía, programación en parejas, las historias, el ciclo semanal, el ciclo trimestral, flujos de trabajo, construcción de 10 minutos, integración continua, prueba de programación y diseño incremental. También hay 11 prácticas como corolario (Beck & Andres, 2004), esta versión extendida de XP fue introducida en 2004 por Beck y otros, sin embargo, muy pocas investigaciones se han centrado en la nueva versión, esto podría ser simplemente debido al hecho de que cualquier método lleva tiempo para ganar fuerza y popularidad, y que sólo puede ser una cuestión de tiempo antes de que el uso de la versión revisada alcance los mismos niveles que el uso de la original.

Extreme Programing es descrita por Beck como "*...una metodología ligera para pequeños y medianos equipos de desarrollo de software en la cara de los requerimientos imprecisos o rápidamente cambiantes...*" (Beck & Zapata Martínez, 2002), reconoce explícitamente que XP no es un conjunto de técnicas de desarrollo nuevos y revolucionarios. Más bien, es un conjunto de principios probados y fiables, bien establecidos como parte de la sabiduría convencional de la ingeniería de software, pero llevado a un extremo nivel de ahí el nombre "programación extrema".

Scrum

No se puede hablar de Scrum sin mencionar a Takeuchi y Nonaka (1986), donde presentan un proceso adaptativo, rápido y auto-organizado de desarrollo de productos y exponen por primera vez el término Scrum, que se deriva del mismo término en rugby, que hace referencia a como se devuelve un balón que ha salido fuera del campo, al terreno de juego de una manera colectiva. Scrum surgió como práctica en el desarrollo de productos tecnológicos y no sería hasta 1993 que Jeff Sutherland aplicará el modelo al desarrollo de software en la Easel Corporation, como lo muestran Avison y Fitzgerald (2006).

Scrum se centra en la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. Entonces, el dueño del producto decide qué elementos del backlog se deben desarrollar en el sprint siguiente. Los miembros del equipo coordinan su trabajo en un diario de stand-up de la reunión. Un miembro del equipo, el “Scrum Master” (equivalente al gerente del proyecto), es el encargado de resolver los problemas que impiden que el equipo trabaje eficazmente (Schwaber & Beedle, 2002), este equipo generalmente es de diez o menos componentes, aunque Schwaber y Beedle recomiendan equipos de cinco integrantes (Schwaber & Beedle, 2002), dividiendo el equipo principal en equipos más pequeños si fuera necesario.

Aún con las recomendaciones que presentan Schwaber y Beedle (2002), para los proyectos actuales y más concretamente los desarrollos para dispositivos móviles, el equipo de desarrollo se está encontrando con problemas de requisitos variables y tecnologías con características muy diferentes. En esta situación se recomienda que el primer sprint tenga una funcionalidad implementada con la tecnología que está dando problemas, se seguirá con un product backlog², priorizando las tareas a ejecutarse, con el objetivo de subir la moral a los desarrolladores y a todo el equipo en general.

Test Driven Development (TDD)

Desarrollo orientado a las pruebas, condiciona la mentalidad de los desarrolladores guiándolos a través del desarrollo y enfocándose en la calidad del producto final, según Astels (2003), es un estilo de desarrollo donde se mantiene un juego de pruebas del programador exhaustivo, ninguna parte del código pasa a producción a no ser que pase sus pruebas asociadas, se escriben primero las pruebas y estas determinan el código que se necesita escribir y debido a su radical planteamiento a la hora de escribir los códigos, cambia drásticamente la mentalidad de cualquier

2. Lista de objetivos/requisitos priorizada, representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto («Lista de objetivos / requisitos priorizada (Product Backlog) | proyectos Ágiles», s. f.).

equipo de desarrollo, generalmente agilizando los resultados y aumentando la calidad del sistema.

TDD a veces es entendido como un procedimiento para asegurar la calidad y originalmente fue pensado como una técnica para mejorar la productividad, el aumento de la calidad fue un efecto secundario por esto es que hoy en día podemos encontrar muchas experiencias en las cuales se ha utilizado TDD como parte de Extreme Programming, es más difícil encontrar experiencias en las cuales se documente la utilización de TDD como metodología aislada, normalmente aparece complementando a otra metodología (Schmidkonz & Staader, 2007), (Letelier, Canós, & Sánchez, s. f.).

Test Driven Development es una de las metodologías con mayor acogida en el campo profesional y que continúa expandiéndose debido a sus buenos resultados. La tendencia actual es integrar TDD independientemente en cualquier metodología ya sea ágil («Scrum Alliance - TDD and Scrum», 2011) o tradicional (Letelier *et al.*, s. f.) y aprovechar los beneficios de practicar una metodología que siempre permite deshacer los errores, asegurar una calidad del producto y protegerse de errores tanto malintencionados como humanos.

Metodologías usadas actualmente para el desarrollo de aplicaciones móviles

Una metodología de desarrollo nueva, especialmente diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés Technical Research Centre of Finland) en Finlandia que lideran una corriente muy importante de desarrollo ágil (Abrahamsson, 2007), muy centrada en las plataformas móviles, y que se presenta con más detalle en Mobile-D: an agile approach for mobile application development (Abrahamsson *et al.*, 2004). El método se basa en prácticas ágiles como Extreme Programming y crystal; las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software; según Abrahamsson (Abrahamsson *et al.*, 2004), Mobile-D debe ser utilizado por un equipo de no más de diez desarrolladores, trabajando en conjunto para suministrar un producto listo en un plazo máximo de diez semanas.

Otro punto de vista muy importante en la actualidad es la propuesta de Rahimian y Ramsin (Rahimian & Ramsin, 3) HMD (Hybrid Methodology Design), la cual, se apoya en una combinación del desarrollo adaptativo de software (Adaptive Software Development, ASD) y el diseño de nuevos productos (New Product Development, NPD); parte del ciclo de vida tradicional (análisis, diseño, implementación, pruebas y desarrollo) e incluye además una fase de comercialización.

La más reciente propuesta de metodologías diseñadas específicamente para aplicaciones móviles y que aún se encuentra en etapa experimental se denomina Mobile Development Process Spiral (Nosseir, Flood, Harrison, & Ibrahim, 2012), la cual es un modelo impulsado por la usabilidad y toma como base el modelo espiral.

Metodología

Generalidades del desarrollo de aplicaciones para dispositivos móviles

La investigación se dividió en tres fases, en la primera, se realizó una revisión de las metodologías ágiles para desarrollo de software, así como también, la identificación de los requisitos específicos de las aplicaciones para dispositivos móviles, en especial las que difieren del desarrollo de software tradicional. Posteriormente, se realizó un diagnóstico de pertinencia de las metodologías ágiles frente a los requisitos específicos del desarrollo de aplicaciones para dispositivos móviles.

En la segunda fase, se realizó una caracterización del desarrollo de aplicaciones para dispositivos móviles en la etapa de análisis, diseño e implementación, así como una revisión de las herramientas de gestión de calidad que ofrecen las metodologías ágiles y son susceptibles a ser implementadas en el desarrollo de aplicaciones para dispositivos móviles.

Para finalizar, se identificaron algunas prácticas de las metodologías ágiles, susceptibles a ser aplicadas en el desarrollo de software para aplicaciones móviles, las cuales, proporcionaron la base para estructurar una guía metodológica, orientada a mejorar la calidad de las aplicaciones y el tiempo de desarrollo en las mismas.

Resultados y discusión

Análisis de los requerimientos propios de las aplicaciones para dispositivos móviles

El mercado de las aplicaciones móviles, ha experimentado una rápida expansión durante los últimos años; las plataformas y equipos móviles, siguen mejorando su desempeño; la demanda de los usuarios por una amplia variedad de aplicaciones móviles va en aumento (Restivo, s. f.); sin embargo, el desarrollo de software para las plataformas móviles viene con características y restricciones únicas, presentadas en la tabla 1, que se aplican en la mayoría de las etapas del ciclo de vida, las características más relevantes son identificadas en las investigaciones de Abrahamsson, Satyanarayanan, Barnard y Beck, entre otros (Abrahamsson,

2005), (Abrahamsson et al., 2004), (Satyanarayanan, 1998), (Barnard, Yi, Jacko, & Sears, 2005) tales como: un alto nivel de competitividad, corto tiempo de entrega, movilidad, portabilidad, capacidades específicas y constantemente cambiantes de las terminales, sistemas operativos diferentes e incompatibles, entre otras, lo que provoca que sea un proceso más complejo, Tabla 1.

Tabla 1. Relación de requisitos propios de las aplicaciones para dispositivos móviles.

Requisitos	Descripción
Canal	Disponibilidad, caída de la señal, ancho de banda, diferencia de redes de conectividad (WI-FI, BLUETOOTH, HSPA, EDGE, GSM, UMTS, 3G Y 4G).
Movilidad	Migración de direcciones, cambio de estación base, gestión de la información por localización (GPS).
Portabilidad	Tamaño de las pantallas (resistivas, capacitativas y multitáctiles), variedad de teclado QWERTY o en pantalla, diversidad de dispositivos, sistemas operativos diferentes e incompatibles.
Capacidades de los terminales	Limitación de los recursos físicos: Potencia del procesador, interfaz gráfica, capacidad de almacenamiento, duración de las baterías, especificaciones de hardware rápidamente cambiantes en las terminales.
Corto tiempo de entrega	Industria en constante cambio, alto nivel de competitividad, las aplicaciones se entregan en versiones rápidas.
Usabilidad	Variedad de usuarios (características cognitivas, edad, nivel de madurez y capacidad intelectual, etc.), contextos en los que se utilizará la aplicación móvil, facilidad de aprendizaje, eficiencia, satisfacción y velocidad de utilización.

Construida a partir de: (Abrahamsson, 2005), (Beck & Zapata Martínez, 2002), (Abrahamsson et al., 2004), (Satyanarayanan, 1998) - (Abrahamsson, 2005).

De todo esto se puede decir, que el diseño de sistemas de software móvil es, en gran medida, más complejo que el tradicional, lo que obliga a los desarrolladores a considerar otras formas de mejorar su producción, por esto, el uso de metodologías ágiles es una excelente alternativa para el desarrollo de aplicaciones para dispositivos móviles, gracias a la gran facilidad de adaptación que poseen, sin embargo, las características de estos desarrollos demandan algunos ajustes sobre las actuales metodologías ágiles, los cuales, fueron abordados en el desarrollo de la guía metodológica AEGIS-MD.

Diagnóstico de las metodologías ágiles aplicadas en la guía metodológica AEGIS-MD

El uso de métodos ágiles de desarrollo de software ha recibido tanto apoyo como argumentos opuestos. Los principales argumentos en contra son la falta de validación y una cierta incertidumbre para distinguir los métodos ágiles de programación ad-hoc, sin embargo, algunos autores defienden la teoría de que los métodos ágiles proporcionan un enfoque de desarrollo organizado (Outi Salo & Valtion teknillinen tutkimuskeskus, 2006), según su enfoque y características esenciales, XP por ejemplo, se centra en las mejores prácticas para el desarrollo de software y consta de doce prácticas (Beck & Zapata Martínez, 2002), mientras que SCRUM, se centra en la gestión de proyectos en situaciones en las que es difícil planificar el futuro (Schwaber & Beedle, 2002). Por último TDD está más orientado a las pruebas, por lo que normalmente aparece complementando a otra metodología (Schmidkonz & Staader, 2007), (Letelier *et al.*, s. f.).

La Tabla 2 muestra un paralelo con respecto a criterios relevantes del ciclo de vida de un proyecto software basado en los siguientes parámetros: A. Brinda soporte a la gestión del proyecto, B. Se describe el proceso para la fase indicada, C. Describe prácticas, actividades y/o artefactos para la etapa.

Tabla 2. Comparativa metodologías vs ciclo de vida.

	Especificación de requisitos			Diseño			Codificación			Test unitarios			Test de Integración			Test del sistema			Test de aceptación		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
SCRUM	X	X	X	X			X			X			X	X	X						
TDD					X	X		X	X		X	X		X	X		X	X		X	X
XP		X	X		X	X		X	X		X	X		X	X		X	X		X	X

Construida a partir de: (Abrahamsson et al., 2003) , (Schwaber & Beedle, 2002), («Scrum Alliance - TDD and Scrum», 2011), (Abrahamsson, 2007), (Abrahamsson, Salo, Ronkainen, & Warsta, 2002).

De este análisis se puede identificar que las metodologías ágiles están centradas en diferentes fases del ciclo de vida, TDD por ejemplo, se enfoca en las fases de pruebas, mientras que XP esta más orientada a las prácticas, las dos son por definición ajustables, ya que permiten un uso parcial o total de las mismas y a pesar de no estar orientadas a la calidad, sino a la productividad (Abrahamsson *et al.*, 2002), dichas metodologías, demuestran tener un conjunto de prácticas y métodos eficientes a la hora de conseguir un producto de calidad; mientras que SCRUM por otro lado, apunta más a la gestión de proyectos, pero no presenta

prácticas concretas que aseguren la calidad, su propósito es sacar el proyecto adelante de una forma colectiva y colaborativa (Schwaber & Sutherland, 2011).

Sin embargo, en el desarrollo de aplicaciones para dispositivos móviles, no parece factible utilizarlas como metodologías únicas o individuales, la adopción de prácticas y procesos complementarios entre ellas parece ser una mejor opción a la hora de asegurar la calidad y la agilidad, debido a las características especiales que presentan estos desarrollos (Abrahamsson, 2005), pero cuando se trata de comparar las características de las aplicaciones móviles a las de un método ágil, la dificultad proviene, en parte, del hecho de que los límites de las metodologías ágiles no están claramente establecidos. Una visión global de la investigación en ese campo se presenta en el trabajo de Dyba y Dingsoyr (2009), los resultados indican, que la introducción de los métodos ágiles a proyectos de software de tamaño reducido genera grandes beneficios, especialmente si las prácticas ágiles no sustituyen completamente a las tradicionales, sino que trabajan en conjunto, sin embargo, especialmente en el caso de las aplicaciones móviles, el problema de la identificación del cliente es mucho más complejo (Abrahamsson, 2005).

Mejores prácticas en el desarrollo de software aplicadas en la guía

El ciclo de vida de la guía metodológica ágil AEGIS-MD propuesta, toma los elementos más representativos y con mayor aceptación dentro de los equipos de desarrollo («State of Agile Development Survey Results | VersionOne», s. f.), que ofrecen las metodologías Extreme Programming (XP), SCRUM y Test Driven Development (TDD) y son susceptibles a ser adaptadas al desarrollo de aplicaciones para dispositivos móviles, centrándose en los requisitos específicos de este desarrollo (Abrahamsson, 2005), (Abrahamsson *et al.*, 2004), especialmente en los tiempos de desarrollo cada vez más cortos y la identificación temprana de los usuarios potenciales.

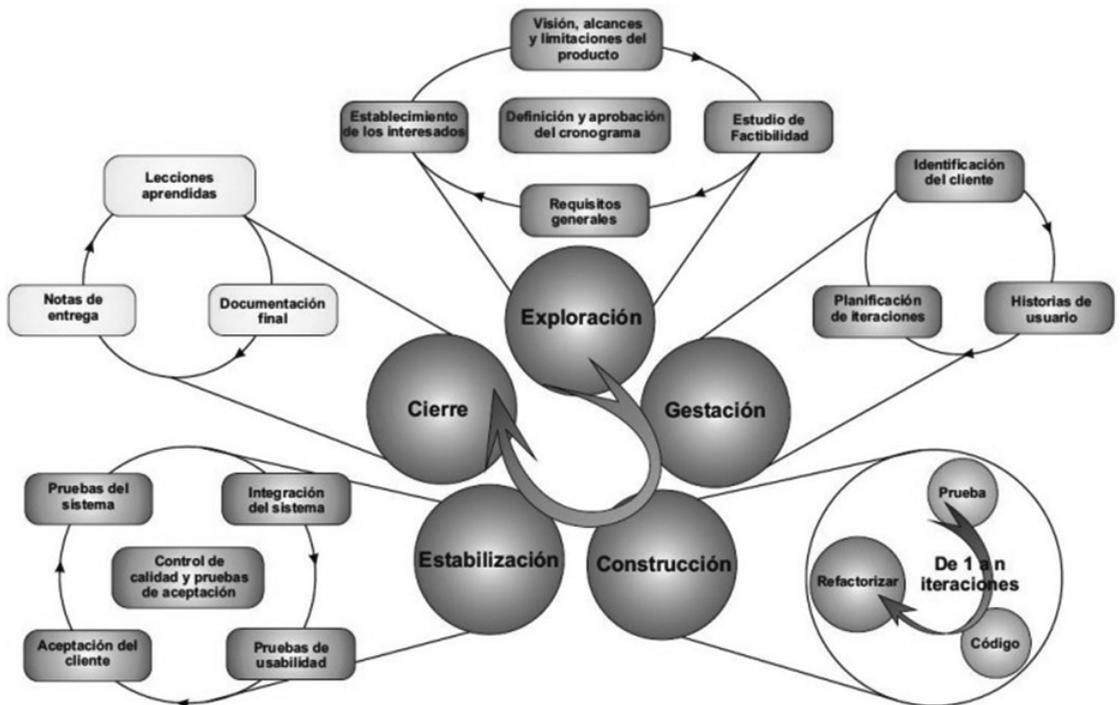
Su enfoque, es el de ser una metodología iterativa y liviana, combinando las que han demostrado ser prácticas óptimas para desarrollar software, ésta propuesta está orientada, a capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones pequeñas para construir un producto bloque a bloque durante todo el ciclo de vida de desarrollo, su principal objetivo, es reducir el tiempo de desarrollo y contribuir con la corrección temprana de errores, generando un menor costo para el proyecto, y finalmente, permitir un mayor control sobre este, disminuyendo posibles riesgos.

La guía metodológica AEGIS-MD se basa en los siguientes principios: La colaboración y la cooperación entre todas las partes interesadas son esenciales, identificación temprana de los usuarios, comunicación y retroalimentación constante, escribir las pruebas primero y un buen estado de ánimo conlleva mejores resultados; también contempla los siguientes roles: Cliente, gerente de proyecto, desarrollador y tester, cabe resaltar que un actor puede ejecutar más de

un rol, especialmente en la fase de construcción, se recomienda que sea realizada en parejas, un programador diseña las pruebas que el otro va a implementar en su código y viceversa; adicionalmente contempla seis (6) herramientas: Estudio de factibilidad, historias de usuario, requisitos generales, identificación de usuarios, notas de aceptación y lecciones aprendidas.

Dado que la guía metodológica AEGIS-MD tiene un enfoque iterativo, el ciclo de vida se divide en cinco (5) fases: exploración, gestación, construcción, estabilización y cierre, Figura 1, cada una de ellas, está dividida en criterios de entradas, actividades y criterios de salida, además, se deben realizar reuniones de los integrantes del equipo en todas las fases del ciclo de vida (Se recomienda una periodicidad mínima de 2 días, con una duración máxima de 10 minutos para cada reunión) con el fin de conocer el estado actual de sus tareas y posibles dificultades que no hayan permitido el avance en el proyecto; y una reunión de cierre (su duración varía entre 1 y 3 horas) para planificar la siguiente fase.

Figura 1. Guía metodológica AEGIS-MD.



Fuente. Autor.

Durante la fase de exploración, se realiza la planificación y establecimiento del proyecto, se determinan la visión, alcances y limitaciones del producto, se realiza el estudio de factibilidad, el establecimiento de los interesados y la definición y aprobación del cronograma de trabajo con el cliente, Figura 2.

Figura 2. Procesos de la fase de exploración.



Entradas:

- Propuesta del producto.
- Contrato.
- Documento de requisitos generales.
- Estudio de factibilidad.
- Documento del cronograma.

Fuente. Autor.

Durante la fase de gestación se realiza la identificación y caracterización de los usuarios potenciales del producto, la recolección de las historias de usuario, priorizando en las tareas y los contextos en los que se utilizará la aplicación móvil y sus atributos de usabilidad, a partir de las tarjetas de historias de usuarios, además, se documenta y acuerda con el cliente la lista de objetivos/ requisitos priorizada (product backlog) así como, la planificación y priorización de las iteraciones, Figura 3.

Figura 3. Procesos de la fase de gestación.



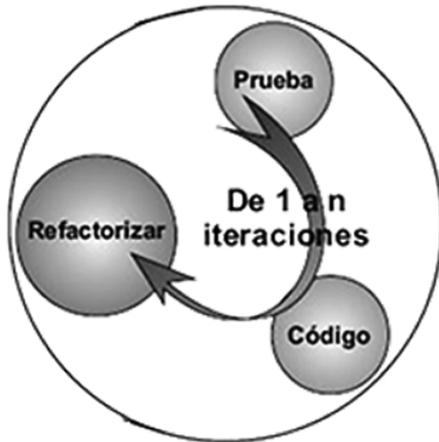
Entradas:

- Formato de identificación de usuarios.
- Historias de usuario.
- Planificación de iteraciones (product backlog).

Fuente. Autor.

Durante la fase de construcción se debe desarrollar el software, cumpliendo con las necesidades planteadas por el usuario en las historias de usuario, mediante un ciclo de desarrollo iterativo e incremental, basado en el algoritmo TDD (Astels, 2003), el cual, consta de tres pasos: escribir la especificación del requisito, implementar el código y refactorizar para eliminar duplicidad y hacer mejoras, Figura 4.

Figura 4. Procesos de la fase de construcción.



Entradas:

- Los requisitos del proyecto han sido identificados.
- Las historias de usuario han sido diligenciadas y priorizadas.
- El equipo se ha reunido y ha sido entrenado en el método de desarrollo.
- El entorno de desarrollo ha sido establecido.

Fuente. Autor.

Durante la fase de estabilización el objetivo es realizar una versión totalmente funcional del sistema en desarrollo, mediante el control de calidad, las tareas se dividen en integración del sistema, pruebas de usabilidad (Nielsen, 1994), aceptación del cliente (Claudia Verónica Ruata & Juan Palacio, 2007) y pruebas del sistema (Beck & Zapata Martínez, 2002); esta fase se activa al terminar la primera iteración y se mantiene activa durante toda la fase de construcción, Figura 5.

Figura 5. Procesos de la fase de estabilización



Entradas:

- Iteraciones implementadas.
- Notas de aceptación.
- Informe de pruebas.

Fuente. Autor.

Para finalizar, la fase de cierre tiene por objetivo realizar la documentación final del sistema, notas de entrega y el análisis de todas las actividades que se realizaron, se contemplan las habilidades y debilidades del equipo, con el fin de mejorarlas; además, se revisan las lecciones aprendidas, las cuales se deben tener en cuenta para próximos proyectos. La etapa culmina en la liberación final de todo el software, Figura 6.

Figura 6. Procesos de la fase de cierre.



Fuente. Autor.

Conclusiones

Este trabajo aporta a la comunidad académica, productiva e investigativa, interesada en el desarrollo de aplicaciones para dispositivos móviles, una guía cimentada en características de las metodologías ágiles XP, SCRUM y TDD, tales como: reducción en los tiempos de entrega, en la proyección de las iteraciones y la maduración del proyecto; además del aumento en la flexibilidad de las diferentes etapas del ciclo de vida de una aplicación.

La guía metodológica AEGIS-MD, pretende convertirse en un referente para la incorporación de metodologías, no solo en el ámbito de las aplicaciones para dispositivos móviles, sino también, en desarrollos de tamaño reducido, con cortos tiempos de entrega, que utilicen otro tipo de tecnologías.

AEGIS-MD, por estar basada en metodologías ágiles y acorde con la principal característica de estas, “adaptabilidad”, también podría ser adecuada a cualquier tipo de desarrollo de software de tamaño reducido, como lo es una aplicación web o una aplicación pequeña de escritorio, que pueda ser desarrollada por un equipo de tamaño reducido.

Referencias bibliográficas

Abrahamsson, P. 2005. *Keynote: Mobile software development—the business opportunity of today*. En: *Proceedings of the International Conference on Software Development* (pp. 20-23). Citeseer.

Abrahamsson, P. 2007. *Agile software development of mobile information systems*. En: *Proceedings of the 19th international conference on Advanced information systems engineering* (pp. 1-4). Springer-Verlag.

Abrahamsson, P. Hanhineva, A. Hulkko, H. Ihme, T. Jääliñoja, J. Korkala, M. Salo, O. 2004. *Mobile-D: an agile approach for mobile application development*. En: *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (pp. 174-175). ACM.

Abrahamsson, P. Salo, O. Ronkainen, J. Warsta, J. 2002. *Agile software development methods: Review and analysis*. VTT Finland.

Abrahamsson, P. Warsta, J. Siponen, M. Ronkainen, J. 2003. *New directions on agile methods: a comparative analysis* (pp. 244-254). IEEE. <http://doi.org/10.1109/ICSE.2003.1201204>

Astels, D. 2003. *Test-driven development : a practical guide*. Upper Saddle River, N.J.; London: Prentice Hall PTR.

Avison, D. Fitzgerald, G. 2006. *Information system development*. Maidenhead: McGraw-Hill Education.

Barnard, L. Yi, J. Jacko, J. Sears, A. 2005. *An empirical comparison of use-in-motion evaluation scenarios for mobile computing devices*. *International Journal of Human-Computer Studies*, 62(4), 487-520.

Beck, K. Andres, C. 2004. *Extreme programming explained : embrace change*. Boston, MA: Addison-Wesley.

Beck, K. Zapata, J. 2002. *Una Explicación de la programación extrema. Aceptar el cambio*. Madrid [etc.]: Addison Wesley.

Dyba, T. Dingsoyr, T. 2009. *What Do We Know about Agile Software Development?* IEEE Software, 26(5), 6-9. <http://doi.org/10.1109/MS.2009.145>

Letelier, P. Canós, J. Sánchez, E. (s. f.) *An Experiment Working with RUP and XP*. En M. Marchesi & G. Succi (Eds.), *Extreme Programming and Agile Processes in Software Engineering* (Vol. 2675, pp. 41-46). Berlin, Heidelberg: Springer Berlin Heidelberg. Recuperado a partir de http://www.springerlink.com/index/10.1007/3-540-44870-5_6

Lista de objetivos / requisitos priorizada (Product Backlog) | proyectos Ágiles. (s. f.). Recuperado 20 de junio de 2013, a partir de <http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>

Manifiesto por el Desarrollo Ágil de Software. (s. f.). Recuperado 13 de noviembre de 2012, a partir de <http://www.agilemanifesto.org/iso/es/>

Nielsen, J. 1994. *Usability engineering*. San Francisco, Calif.: Morgan Kaufmann Publishers. Recuperado a partir de <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=582280>

Nosseir, A. Flood, D. Harrison, R. Ibrahim, O. 2012. *Mobile Development Process Spiral* (pp. 281-286). IEEE. <http://doi.org/10.1109/ICCES.2012.6408529>

Rahimian, V. Ramsin, R. 2008. *Designing an agile methodology for mobile software development: A hybrid method engineering approach*. *Research Challenges in Information Science*. RCIS 2008. Second International Conference on, 337-342. <http://doi.org/10.1109/RCIS.2008.4632123>

Restivo, K. (s. f.) *Worldwide Quarterly Mobile Phone Tracker*. Recuperado 9 de julio de 2013, a partir de http://www.idc.com/tracker/showproductinfo.jsp?prod_id=37

Ruata, C. Palacio, J. 2007. *Scrum Manager Gestión de Proyectos*. Recuperado 13 de septiembre de 2013, a partir de <http://www.safecreative.org/?wicket:interface=:0:::>

Salo, O. Abrahamsson, P. 2008. *Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum*. *IET Software*. 2(1), 58. <http://doi.org/10.1049/iet-sen:20070038>

Salo, O. Valtion, T. 2006. *Enabling software process improvement in agile software development teams and organisations*. VTT Technical Research Centre of Finland, [Espoo, Finland].

Satyanarayanan, M. 1996. *Fundamental challenges in mobile computing*. En: *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing* (pp. 1-7). ACM.

Satyanarayanan, M. 1998. *Fundamental Challenges in Mobile Computing*.

Schmidkonz, C. Staader, J. 2007. *Piloting of Test Driven Development in Combination with Scrum*. Recuperado 18 de julio de 2013, a partir de <http://members.scrumalliance.org/resources/267>

Schwaber, K. Beedle, M. 2002. *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.

Schwaber, K. Sutherland, J. 2011. *The scrum guide*.

Scrum Alliance - TDD and Scrum. 2011. Recuperado 13 de noviembre de 2012, a partir de <http://www.scrumalliance.org/resources/267>

State of Agile Development Survey Results | VersionOne. (s. f.). Recuperado 10 de julio de 2013, a partir de http://www.versionone.com/state_of_agile_development_survey/2011/

Szalvay, V. 2004. *An introduction to agile software development*. Danube Technologies.

Takeuchi, H. Nonaka, I. 1986. *The new new product development game*. Harvard Business Review.