

Metodologías ágiles y desarrollo de aplicaciones móviles

Yohn Daniel Amaya Balaguera

yohn.amaya@uptc.edu.co

Universidad Pedagógica y Tecnológica de Colombia, Colombia
Avenida Central del Norte 39-115
Tunja, Boyacá, Colombia
Director
Jorge Enrique Otálora Luna
Magister en Ingeniería

Resumen: *Las metodologías ágiles han ganado popularidad desde hace algunos años, ya que constituyen una buena alternativa para proyectos a corto plazo, en especial, aquellos proyectos en donde los requisitos están cambiando constantemente. Un ejemplo de esto, son las aplicaciones para dispositivos móviles, debido a que éstas tienen que satisfacer una serie de características especiales que difieren del software tradicional, pero el uso de metodologías no se considera importante en este ámbito. En este trabajo, se exponen algunas características de las metodologías ágiles más usadas, se presentan algunas generalidades del desarrollo de aplicaciones para dispositivos móviles y sistemas operativos, para concluir con una síntesis de metodologías usadas actualmente para aplicaciones móviles.*

Palabras clave: Metodología, ágil, aplicación, móvil.

Abstract: *Agile methodologies have gained popularity in recent years as they are a good alternative for short-term projects, especially those projects where the requirements are constantly changing, an example of this, are mobile applications because that you have to meet a number of special features that differ from traditional software, but the use of methodologies is not considered important in this area, in this paper we describe some features of the most used agile methodologies, are some generalities development applications for mobile devices and operating systems, concluding with a synthesis of methodologies currently used for mobile applications.*

Keywords: Methodology, agile, application, mobile.

1 Introducción

La ingeniería de software juega un papel importante en el desarrollo, portabilidad, mantenibilidad, funcionalidad, fiabilidad y productividad del software y junto con ella, las metodologías ágiles, las cuales son procesos para desarrollar software de manera rápida con gran facilidad de adopción por los equipos de trabajo [1]. Su origen se remonta a mediados de la década de los 1990 como una reacción a las metodologías tradicionales, pero reconocidas formalmente, luego de una reunión de miembros prominentes de la comunidad científica efectuada del 11 al 13 de febrero de 2001, en Snowbird Utah donde nace el término “métodos ágiles” [2], término que implica ser eficaz y fácil de manejar [3]. Sin embargo, la aparición de las metodologías ágiles no pueden ser asociadas a una única causa, sino a todo un conjunto de ellas. Los factores que comúnmente se mencionan son la pesadez, lentitud de reacción y exceso de documentación [4] [5], en definitiva, falta de agilidad de los modelos de desarrollo formales. Otro punto importante sería la explosión de la red, las aplicaciones Web y las aplicaciones móviles, así como el crecimiento notorio del movimiento open source, pero cada autor determina el surgimiento de las metodologías ágiles de diferentes maneras.

A todo esto se puede añadir un cambio bastante importante, en cuanto a la demanda del mercado del software cada vez más orientada a la Web y a dispositivos móviles, con requisitos muy volátiles y en constante cambio, como canal, movilidad, portabilidad, capacidades específicas de las terminales, entre otras [6] [7], que

requieren tiempos de desarrollo cada vez más cortos, lo que provocó que las empresas se fijaran más en nuevos desarrolladores, con nuevos métodos “amateurs” que se combinan con técnicas de las metodologías formales o bien los modelos de desarrollo de las comunidades open source, lo que es muy conveniente para el tamaño y agilidad de los proyectos de aplicaciones móviles, las cuales han venido creciendo considerablemente en los últimos años, en número de desarrollos, pero no en calidad. Esto nos permite reconocer que los métodos de software deben ser adaptados [5] e involucrados en el desarrollo de aplicaciones para dispositivos móviles, si es que se quiere lograr un efecto óptimo en este ámbito [8].

Sin embargo, existen muy pocas investigaciones hasta el momento sobre el uso de los métodos ágiles en el desarrollo de aplicaciones para dispositivos móviles, ya que, cuando se trata de acoplar las características de las aplicaciones móviles con un método ágil, la dificultad proviene, en parte, del hecho de que los límites de las metodologías ágiles no están claramente establecidos. Igualmente, el uso de métodos ágiles de desarrollo de software ha recibido tanto apoyo, como argumentos opuestos, el principal argumento en contra es la falta de validación, sin embargo, algunos autores defienden la teoría de que los métodos ágiles proporcionan un enfoque de desarrollo organizado [9]. Una visión global de la investigación en ese campo se presenta en [10] donde los autores dividen su investigación en cuatro categorías: introducción y adaptación, factores humanos y sociales, la percepción de los métodos ágiles, y estudios comparativos. 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.

En otro punto de vista, Abrahamsson [6] realiza una comparación directa entre las características del método ágil y características de las aplicaciones móviles, centrándose en la cantidad de documentación producida, la planificación involucrada, el tamaño del equipo de desarrollo, la identificación del cliente, y la orientación a objetos. Excepto la identificación del cliente, todas las características de las metodologías ágiles son adecuadas para el desarrollo de aplicaciones móviles. El cliente puede ser identificado como el distribuidor de software, sin embargo, especialmente en el caso de las aplicaciones móviles, el problema de la identificación del cliente es mucho más complejo.

Debido a esto, esta investigación presenta una síntesis sobre el desarrollo de aplicaciones y metodologías de desarrollo ágil, para lo cual, se han seleccionado las dos (2) metodologías ágiles más referenciadas, con mayor presencia de documentación en internet y orientadas a desarrollos de tamaño reducido propio de las aplicaciones para dispositivos móviles, según un estudio realizado en 2013[11], como son Scrum, Extreme Programming (XP), además presenta una síntesis de metodologías usadas actualmente para aplicaciones móviles. En la primera parte, se exponen algunas características de las metodologías ágiles más usadas; en la segunda, se expondrán algunas generalidades del desarrollo de aplicaciones para dispositivos móviles, para concluir con una revisión de las metodologías utilizadas actualmente en el desarrollo de aplicaciones para dispositivos móviles.

2 Fundamento teórico

El presente capítulo muestra una breve descripción sobre metodología del software y metodologías ágiles con buena aceptación dentro de los equipos de desarrollo [1] [4] [12] y adaptables a las características específicas del desarrollo de aplicaciones para dispositivos móviles.

2.1 Metodología de desarrollo

Avison y Fitzgerald presentan una descripción de las metodologías de desarrollo y destacan sus principales componentes, fases, herramientas y técnicas [13]. Sin embargo, una metodología es algo más que una colección, puesto que se basa en una filosofía, distinguiéndose de los métodos o de las simples recetas que marcan unos pasos a seguir y ya está. Así, las metodologías difieren ya sea por la cantidad de fases, las técnicas de cada fase, el contenido de la fase o en su base filosófica, todo esto se aplica, dependiendo del contexto de desarrollo, tamaño del proyecto o del equipo de trabajo, cultura organizacional, entre otros aspectos, por lo que en el caso de los desarrollos móviles, es de vital importancia su selección, para garantizar un producto de calidad.

2.2 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" [14], y 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, e incremental donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto. Las iteraciones individuales deben producir alguna característica completamente funcional o mejorada [15], su principal objetivo es reducir el tiempo de desarrollo, del mismo modo que con el modelo en cascada o waterfall que fuera introducido por Royce en 1970 [16] y usado inicialmente para desarrollo de software, pero expandido por Boehm en 1981 [17], donde todos los requisitos se analizan antes de empezar a desarrollar, sin embargo, los requisitos se dividen en “incrementos” independientemente funcionales.

Muchas ideas que se plantean en las metodologías ágiles no son nuevas. Gran parte de ellas ya fueron reflejadas por Brooks en su libro, The Mythical Man Month [18] y en gran parte responden al sentido común. Algunos autores consideran que se ha cumplido un círculo que empezó con una reacción provocada por múltiples factores y señalada temporalmente por el manifiesto de Dijkstra, en el cual se hacía un llamamiento a la disciplina y que se cierra con el ya famoso Manifest for Agile Software Development, una petición por la relajación de los procesos en pro de las personas [14].

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, lentitud de reacción y exceso de documentación, en definitiva, falta de agilidad de los modelos de desarrollo formales. Otro punto importante sería la explosión de la red, las aplicaciones Web y las aplicaciones móviles, así como el crecimiento notorio del movimiento open source, en un estudio realizado en 2012 por la empresa Version One [11], donde entrevistan a desarrolladores de diferentes compañías que han hecho uso de las metodologías ágiles, sobre las principales razones para la adopción de estas prácticas, como se muestra en la figura 1, la principal razón, es acelerar el tiempo de entrega con el 38% de los encuestados diciendo que es muy importante y un 39% altamente importante.

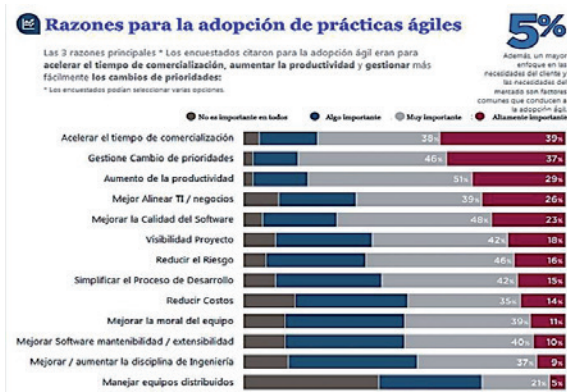


Figura 1. Razones para la adopción de prácticas ágiles, adaptada de [12].

A todo esto se puede añadir un cambio bastante importante, en cuanto a la demanda del mercado del software, cada vez más orientada a la Web y a dispositivos móviles, con requisitos muy volátiles y en constante cambio, que requieren tiempos de desarrollo cada vez más cortos, lo que provocó que las empresas se fijaran más en nuevos desarrolladores, con nuevos métodos “amateurs” que se combinan con técnicas de las metodologías formales. Los modelos de desarrollo de las comunidades open source pudieron ciertamente determinar la aparición de las metodologías ágiles, pero cada autor determina el surgimiento de las metodologías ágiles de diferentes maneras.

2.2.1 Extreme Programming (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 [19]. 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 corolario [20]. Esta versión extendida de XP fue introducida en 2004 por Beck y otros, sin embargo, muy pocas investigaciones se ha centrado en la nueva versión. Estos 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 del uso de la versión revisada alcanza los mismos niveles que el uso de la original.

Extreme Programming 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...” [19], 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”.

La mayoría de la literatura existente sobre XP sugirieron que sus métodos pueden ser adaptados con facilidad [21] [22]. Sin embargo, Conboy [5] presenta un estudio muy detallado, que propone a XP como una serie de piezas de rompecabezas que encajan como un reloj suizo, a pesar del hecho de que XP es supuestamente adaptable a una amplia variedad de proyectos. Algunas prácticas aunque no agregan valor, son imposibles de eliminar, ya que son necesarias para mantener las otras en su lugar.

2.2.2 Scrum. No se puede hablar de Scrum sin mencionar a Takeuchi y Nonaka [23] 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 y hace referencia a cómo 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 [13].

Scrum se centra 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 [24]. Este equipo generalmente es de diez o menos componentes, aunque Schwaber and Beedle recomiendan equipos de cinco integrantes [24], dividiendo el equipo principal en equipos más pequeños si fuera necesario.

Aun con las recomendaciones que presentan Schwaber y Beedle [24], 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 backlog¹⁵, priorizando las tareas a ejecutarse, con el objetivo de subir la moral a los desarrolladores y a todo el equipo en general.

Scrum es una metodología que goza de gran popularidad y de la cual podemos encontrar bastantes empresas que gustan de utilizarla. Entre ellas podemos encontrar empresas tan importantes como Yahoo o Google. En el

¹⁵ Lista de objetivos/requisitos priorizada, representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto [25]

año 2000 Rising and Janoff publicaron tres proyectos exitosos utilizando Scrum [26].

2.3 Generalidades del desarrollo de aplicaciones para dispositivos móviles

El mercado de aplicaciones móviles ha experimentado una rápida expansión durante los últimos 10 años, las plataformas móviles siguen mejorando su desempeño, y la necesidad de los usuarios de una amplia variedad de aplicaciones móviles va en aumento [27]. El desarrollo de software para las plataformas móviles viene con características únicas y restricciones que se aplican a la mayoría de las etapas del ciclo de vida. Las características distintivas más importantes son identificadas en [28] [29] [30], 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.

Abrahamsson [30], presenta una serie de características a tener en cuenta a la hora de desarrollar software para dispositivos móviles, como son, que el software es liberado en un ambiente incierto y dinámico con un alto nivel de competencia, los equipos que desarrollan aplicaciones móviles son generalmente pequeñas y medias empresas. Las aplicaciones en sí son de pequeño tamaño, se entregan en versiones rápidas con el fin de satisfacer las demandas del mercado y se dirigen a un gran número de usuarios finales. El autor sugiere que en el desarrollo de aplicaciones de software para dispositivos móviles los equipos de desarrollo deben hacer frente al desafío de un entorno dinámico, con modificaciones frecuentes en las necesidades y expectativas del cliente [31], por tanto, deben ser desarrollados con enfoques orientados a ciclos de desarrollo relativamente cortos, propios de las metodologías ágiles para el desarrollo de software.

2.3.1 Características del desarrollo para dispositivos móviles. El desarrollo de aplicaciones móviles tiene que satisfacer una serie de requisitos y condiciones especiales [32] [33] [6] lo que provoca que sea un proceso más complejo. Las características más relevantes, se describen en la tabla 1.

Tabla 1. Relación de requisitos propios de las aplicaciones para dispositivos móviles, (Construida a partir del contenido referenciado).

Requisitos	Descripción
Canal	Disponibilidad, caída de la señal, ancho de banda, la 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, capacitivas y multitáctil), del teclado QWERTY o en pantalla, diversidad de dispositivos, sistemas operativos diferentes e incompatibles.
Capacidades de los terminales	Potencia del procesador, interfaces gráfica, capacidad de almacenamiento, duración de las baterías.

Tiempo	Industria en constante cambio, alto nivel de competitividad, corto tiempo de entrega.
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.

Se puede decir que el diseño de sistemas de software móvil es, en gran media, 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 son 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.

2.3.2 Sistemas operativos para dispositivos móviles. Los sistemas operativos con mayor influencia en el mercado y con un número importante de desarrolladores interesados en participar en las “App Stores”, según un estudio realizado [34] en el año 2012 y presentado en la figura 1, muestra que Android ha encabezado el interés de los desarrolladores con un 77% junto con IOS que posee el 66%, mientras que BlackBerry posee un 34% frente a un 37% de Windows phone.

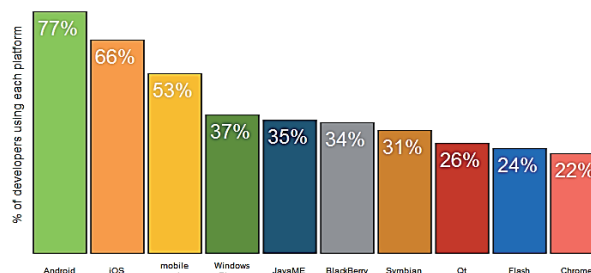


Figura 2. Principales plataformas para desarrolladores Mindshare¹⁶, Tomada de [34].

Android ha demostrado ser un negocio muy beneficioso, ya que sólo en el tercer trimestre de 2012 se comercializaron 136 millones de aparatos con este software y a nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2012 [36] [37], mientras que su principal competidor, el sistema operativo IOS, amplió la brecha con Android tomando solo un 23.6% del mercado, frente a un 51,3% de Android [38], pero en 1er trimestre de 2013 Apple vendió 37,4 millones de teléfonos inteligentes en todo el planeta, aumentando un modesto 7% respecto al año anterior [39].

Por otro lado, la competencia entre Blackberry y Windows Phone se comienza a ver más apretada, ya que las ventas de la empresa Blackberry en el cuarto trimestre de 2012 se redujeron un 44,4% frente al año anterior, tomando solo un 8,8% de presencia en el mercado [38], mientras que para Windows Phone la cuota de mercado

¹⁶ En marketing, se conoce como mindshare (presencia mental) a un parámetro que mide la cantidad o porcentaje de personas que piensan en una determinada marca o producto, cuando se les menciona una categoría o tipo de productos [35].

mundial se ha reducido drásticamente a un 1,8% durante el año pasado [38].

3 Metodologías 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 [31] muy centrada en las plataformas móviles, y que se presenta con más detalle en [28]. 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 [28] 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 [40] 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 incluyen 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 [41], el cual es un modelo impulsado por la usabilidad y toma como base el modelo espiral.

3.1 Mobile-D

Consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas. Las especificaciones completas del método están disponibles en [42].



Figura 3. Ciclo de desarrollo Mobile-D, adaptada de [42].

En la primera fase, explorar, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento actores, definición del alcance y el establecimiento de proyectos. Las tareas asociadas a esta fase incluyen el

establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos.

En la siguiente fase, iniciación, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico, como los recursos físicos, tecnológicos y de comunicaciones (incluyendo el entrenamiento del equipo de desarrollo). Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

En la fase de producción, se repite la programación de tres días (planificación, trabajo, liberación), se repite iterativamente hasta implementar todas las funcionalidades. Primero, se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de "producción", aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente, se puede considerar en esta fase la producción de documentación.

La última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

3.2 Hybrid Methodology Design



Figura 4. Ciclo de desarrollo Hybrid Methodology Design, [40]

Esta metodología utiliza el modelo iterativo incremental para el proceso de desarrollo y así lograr la rápida entrega de software y mejorar las capacidades de gestión de riesgos. Algunas de las características ágiles que se destacan y que también se alinean con las necesidades de desarrollo de aplicaciones móviles, son según [43]:

- Desarrollo basado en pruebas.
- Participación continua del cliente.
- Establecimiento de prioridades en los requisitos.
- Comunicación efectiva.
- Calidad garantizada.
- Desarrolladores expertos.

- Revisión de todo el proceso y sesiones de aprendizaje.
- Proceso de adaptación.

Esta metodología parte del ciclo de vida tradicional y sus especificaciones están disponibles en [40]. La primera iteración se divide la fase de análisis con la intención de mitigar riesgos de desarrollo. De la misma forma, el diseño también se segmenta para introducir algo de diseño basado en arquitectura. La implementación y las pruebas sin embargo se fusionan introduciendo conceptos de desarrollo orientado a pruebas (Test-Driven Development, TDD).

Aparece, además, una fase de comercialización hacia el desarrollo de producto que se imponen en el escenario del desarrollo de aplicaciones para plataformas móviles. Desde el punto de vista metodológico, los autores afirman haberse apoyado en metamodelos, como SPEM (Software Processes Engineering Metamodel, soportado por el entorno de desarrollo de Eclipse) y OPF, (Open Processes Framework), así como en conceptos genéricos de ciclos de vida orientados a objetos como OOSP (Object-Oriented Software Processes).

La segunda iteración realiza una integración de ciertas partes de los modelos NPD (New Product Development), añadiendo la generación de ideas en el inicio del ciclo y una prueba de mercado antes de lanzar la fase de comercialización. La tercera iteración integra directamente el "motor de desarrollo" de los métodos de desarrollo adaptativo (ASD) muy orientados al aseguramiento de la calidad en los procesos de desarrollo con la idea de disponer de la arquitectura física en una fase temprana del proceso. En la cuarta iteración se añaden elementos de prototipado; se refina, además, la fase de iniciación del proyecto, sobre la base del mismo elemento de los procesos adaptativos.

3.3 Mobile Development Process Spiral

Esta propuesta metodológica utiliza el modelo de desarrollo en espiral como base, e incorpora procesos de evaluación de la usabilidad, priorizando la participación del usuario en todos los procesos del ciclo de vida de diseño con el fin de garantizar un diseño centrado en el usuario, aun cuando se trata de un modelo de proceso orientado a proyectos grandes y costosos, ya que está destinado a ser un modelo de reducción de riesgos [41].

El proceso permite, a los desarrolladores de aplicaciones móviles, detallar los criterios de usabilidad de la aplicación. El primer paso es identificar a los usuarios, las tareas y los contextos en los que se utilizará la aplicación móvil. El siguiente paso es dar prioridad a los atributos de usabilidad, identificar qué atributos son los más importantes para la aplicación, y para cada uno definir un conjunto de métricas para verificar el grado en que se cumplen en la aplicación final.

El proceso de desarrollo de aplicaciones móviles en espiral contempla cinco (5) iteraciones, para cada una de ellas tres (3) tareas (determinación de requisitos, diseño y prueba) y finaliza cada iteración con la planificación de la siguiente. En la primera iteración se determinan los requisitos del sistema y se identifican usuarios, tareas y

contextos en los que se utilizará la aplicación. Luego, se definen y priorizan los atributos de facilidad de uso y se identifican métricas para cada atributo; se dibuja un prototipo de la interfaz de aplicación y se realiza la prueba del prototipo, los desarrolladores podrán utilizar diferentes técnicas de usabilidad para medir el valor de cada atributo.

En la segunda iteración, el equipo de desarrollo recogerá más datos y requisitos, explorará si hay más usuarios potenciales, tareas y contextos en los que se utilizará la aplicación. A continuación, los atributos de usabilidad se redefinen y son priorizados, como resultado, los desarrolladores alterarán las métricas para acomodar los requisitos añadidos. En el diseño se realiza un prototipo de alta fidelidad de la interfaz y se realizan las pruebas, utilizando técnicas de usabilidad para cada atributo, la calificación se compara con los resultados de la iteración anterior.

En la tercera iteración, los desarrolladores pueden identificar y priorizar los atributos de usabilidad con mayor claridad utilizando los resultados de la iteración anterior; se desarrolla el diseño de todo el sistema y se realiza la versión alfa con sus respectivas pruebas, el equipo de desarrollo compara los resultados con la calificación de la iteración anterior.

En la cuarta iteración, los resultados de la iteración anterior son utilizados para identificar y dar prioridad a los atributos de facilidad de uso; se desarrolla la versión beta y se libera para su evaluación por parte del cliente.

En la quinta iteración, se desarrolla el producto final; se realiza una evaluación de facilidad de uso, la calificación de cada atributo, se calcula y se compara con la calificación de la fase anterior. Una alteración en el producto final se realiza sobre la base de los resultados y se libera al producto.

4 Conclusiones

Las metodologías ágiles son una excelente alternativa para guiar proyectos de desarrollo de software de tamaño reducido. No obstante, aun cuando existe una gran variedad de ellas, en el ámbito del desarrollo de aplicaciones para dispositivos móviles, su uso no se considera importante y las empresas que utilizan métodos de desarrollo, orientan sus esfuerzos al desarrollo de nuevas metodologías basadas en prácticas de las tradicionales, con algunos aditamentos de técnicas modernas como lo es la usabilidad, sin embargo, las metodologías ágiles necesitan ser adaptadas a las características especiales de estos dispositivos con el fin de obtener productos de calidad.

Dentro de las características se puede mencionar la gran rapidez con que van cambiando las versiones de los sistemas operativos (SO) móviles, la aparición de nuevas prestaciones de hardware, la reñida competencia de las empresas y comunidades de desarrollo por apropiarse del mercado, lo que conduce a la tendencia de desarrollar aplicaciones en cortos lapsos y en la mayoría de casos sacrificando la calidad del producto, precisamente por no seguir una metodología o técnicas de desarrollo

apropiadas y es por eso que en la mayoría de los casos, sus esfuerzos terminan por no dar los frutos esperados.

Referencias bibliográficas

- [1] O. Salo y P. Abrahamsson, «Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum», *IET Softw.*, vol. 2, n.º 1, p. 58, 2008.
- [2] «History: The Agile Manifesto». [En línea]. Disponible en: <http://agilemanifesto.org/history.html>. [Accedido: 11-jul-2013].
- [3] P. Letelier, J. H. Canós, y C. Penadés, «Metodologías Ágiles en el Desarrollo de Software», presentado en VIII Jornadas de Ingeniería del Software y Bases de Datos JISBD, Alicante - España, 2003, pp. 1-8.
- [4] P. Abrahamsson, J. Warsta, M. T. Siponen, y J. Ronkainen, «New directions on agile methods: a comparative analysis», 2003, pp. 244-254.
- [5] K. Conboy y B. Fitzgerald, «Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion», *ACM Trans. Softw. Eng. Methodol.*, vol. 20, n.º 1, pp. 2:1-2:30, jun. 2010.
- [6] P. Abrahamsson, «Keynote: Mobile software development—the business opportunity of today», en *Proceedings of the International Conference on Software Development*, 2005, pp. 20-23.
- [7] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, y O. Salo, «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*, 2004, pp. 174-175.
- [8] V. E. Jyothi y K. Nageswara Rao, «Effective Implementation of Agile Practices - Incoordination with Lean Kanban.», *Int. J. Comput. Sci. Eng.*, vol. 4, n.º 1, pp. 87-91, ene. 2012.
- [9] O. Salo, «Enabling software process improvement in agile software development teams and organisations», *VTT Publ.*, vol. 618, 2006.
- [10] T. Dyba y T. Dingsoyr, «What do we know about agile software development?», *Softw. IEEE*, vol. 26, n.º 5, pp. 6-9, 2009.
- [11] «Annual State of Agile Development Survey Results | VersionOne». [En línea]. Disponible en: <http://www.versionone.com/state-of-agile-survey-results/>. [Accedido: 13-jul-2013].
- [12] «State of Agile Development Survey Results | VersionOne». [En línea]. Disponible en: http://www.versionone.com/state_of_agile_development_survey/2011/. [Accedido: 10-jul-2013].
- [13] D. E. Avison y G. Fitzgerald, *Information system development*. Maidenhead: McGraw-Hill Education, 2006.
- [14] «Manifiesto por el Desarrollo Ágil de Software». [En línea]. Disponible en: <http://www.agilemanifesto.org/iso/es/>. [Accedido: 13-nov-2012].
- [15] V. Szalvay, «An introduction to agile software development», *Danube Technol.*, 2004.
- [16] W. Royce, «Managing the Development of Large Software Systems: Concepts and Techniques», *WESCON West. Electron. Show Conv.*, 1970.
- [17] B. W. Boehm, *Software engineering economics*. Englewood Cliffs, N.J.: Prentice-Hall, 1981.
- [18] F. P. Brooks, *The mythical man-month: essays on software engineering*. Reading, Mass.: Addison-Wesley Pub. Co., 1995.
- [19] K. Beck y J. Zapata Martínez, *Una Explicación de la programación extrema. Aceptar el cambio*. Madrid [etc.]: Addison Wesley, 2002.
- [20] K. Beck y C. Andres, *Extreme programming explained: embrace change*. Boston, MA: Addison-Wesley, 2004.
- [21] J. Bowers, J. May, E. Melander, M. Baarman, y A. Ayoob, «Tailoring XP for Large System Mission Critical Software Development», vol. 2418, D. Wells y L. Williams, Eds. Springer Berlin / Heidelberg, 2002, pp. 269-301.
- [22] J. Rasmusson, «Introducing xp into greenfield projects: lessons learned», *IEEE Softw.*, vol. 20, n.º 3, pp. 21-28, may 2003.
- [23] H. Takeuchi y I. Nonaka, «The new new product development game», *Harv. Bus. Rev.*, 1986.
- [24] K. Schwaber y M. Beedle, *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [25] «Lista de objetivos / requisitos priorizada (Product Backlog) | proyectos Ágiles». [En línea]. Disponible en: <http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>. [Accedido: 20-jun-2013].
- [26] L. Rising y N. S. Janoff, «The Scrum software development process for small teams», *IEEE Softw.*, vol. 17, n.º 4, pp. 26-32, ago. 2000.
- [27] K. Restivo, «Worldwide Quarterly Mobile Phone Tracker». [En línea]. Disponible en: http://www.idc.com/tracker/showproductinfo.jsp?prod_id=37. [Accedido: 09-jul-2013].
- [28] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, y O. Salo, «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*, 2004, pp. 174-175.
- [29] M. Satyanarayanan, «Fundamental Challenges in Mobile Computing.», oct. 1998.
- [30] P. Abrahamsson, «Keynote: Mobile software development—the business opportunity of today», *Proc. Int. Conf. Softw. Dev.*, pp. 20-23, 2005.
- [31] P. Abrahamsson, «Agile software development of mobile information systems», en *Proceedings of the*

19th international conference on Advanced information systems engineering, 2007, pp. 1-4.

- [32] G. H. Forman y J. Zahorjan, «The challenges of mobile computing», *Computer*, vol. 27, n.º 4, pp. 38-47, abr. 1994.
- [33] L. Barnard, J. S. Yi, J. A. Jacko, y A. Sears, «An empirical comparison of use-in-motion evaluation scenarios for mobile computing devices», *Int. J. Hum.-Comput. Stud.*, vol. 62, n.º 4, pp. 487-520, 2005.
- [34] «VisionMobile - Data Mindshare Index», *Mindshare Index Top platforms being used by developers*, 2012. [En línea]. Disponible en:
<http://visualisations.visionmobile.com/main/shared/1>. [Accedido: 01-may-2013].
- [35] R. de Miguel, «Mindshare acerca el futuro del marketing móvil», *Ipmark Inf. Public. Mark.*, n.º 787, pp. 14-16, 2012.
- [36] L. Goasduff y C. Pettey, «Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth», 2012. [En línea]. Disponible en:
<http://www.gartner.com/it/page.jsp?id=1924314>. [Accedido: 30-abr-2013].
- [37] S. K. Crook, S. D. Drake, y B. Hoffman, «Worldwide Mobile Enterprise Management Software 2012–2016 Forecast and Analysis and 2011 Vendor Shares», 2012. [En línea]. Disponible en:
<http://www.idc.com/getdoc.jsp?containerId=236835>. [Accedido: 30-abr-2013].
- [38] R. van der Meulen y Janessa Rivera, «Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012». [En línea]. Disponible en:
<http://www.gartner.com/newsroom/id/2335616>. [Accedido: 01-may-2013].
- [39] S. Bicheno, «Q1 '13: Apple: iPhone Growth at Lowest Level in History», *Strategy Analytics*, 25-abr-2013. [En línea]. Disponible en:
<http://www.strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=8444>. [Accedido: 08-may-2013].
- [40] V. Rahimian y R. Ramsin, «Designing an agile methodology for mobile software development: A hybrid method engineering approach», *Res. Challenges Inf. Sci. 2008 RCIS 2008 Second Int. Conf.*, pp. 337-342, 3.
- [41] A. Nosseir, D. Flood, R. Harrison, y O. Ibrahim, «Mobile Development Process Spiral», 2012, pp. 281-286.
- [42] «Electronics -AGILE - Agile Software Technologies», 21-nov-2012. [En línea]. Disponible en:
<http://virtual.vtt.fi/virtual/agile/mobiled.html>. [Accedido: 30-abr-2013].
- [43] S. B. Kaleel y S. Harishankar, «Applying Agile Methodology in Mobile Software Engineering:

Android Application Development and its Challenges», 2013.