

Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual

Agile methodologies in the development of applications for mobile devices. present state

Yohn Daniel Amaya Balaguera

Resumen

Las metodologías ágiles han ganado popularidad desde hace algunos años, ya que constituyen una buena solución 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 y condicionantes especiales, tales como: canal, movilidad, portabilidad, capacidades específicas de las terminales, entre otras, y aun cuando existen miles de aplicaciones para dispositivos móviles que corren en diferentes sistemas operativos IOS, Android, BlackBerry y Windows Mobile; éstas llenan las expectativas de los usuarios hasta cierto punto por su escasa calidad en el desarrollo, ya que el uso de metodologías de desarrollo de software no se considera importante en este ámbito, por tanto, los desarrollos para dispositivos móviles, hasta el momento, se han venido realizando, principalmente, de manera desordenada y en la mayoría de los casos por desarrolladores individuales que no aplican métodos de ingeniería de software que garanticen su mantenibilidad y por lo tanto su calidad.

Palabras clave: Metodología ágil, aplicación móvil, Android, Blackberry, IOS y Windows Phone.

Abstract

Ágile methodologies have gained popularity in recent years as they are a good solution for short-term projects, especially those projects where requirements are constantly changing, an example of this is the mobile device applications, because you have to satisfy a number of special features and constraints, such as channel mobility, portability, specific capabilities of the terminals, among others, and even when there are thousands of applications for mobile devices running on different operating systems IOS, Android, BlackBerry and Windows Mobile, you meet the expectations of the users to some extent by its low quality in the development, since the use of software development methodologies is not considered important in this area, therefore, mobile developments, so far have been conducted mainly in a disorderly manner and in most cases by individual developers do not apply software engineering methods to ensure maintainability and therefore quality.

Keywords: Agile methodology, Mobile Application, Android, Blackberry, iOS and Windows Phone.

Recibido: Julio 23 de 2013 **Aprobado:** Noviembre 14 de 2013

Tipo de artículo: Investigación científica y tecnológica terminada.

Afiliación Institucional de los autores: Universidad Pedagógica y Tecnológica de Colombia, sede Tunja.

El autor declara que no tiene conflicto de interés.

Introducción

La ingeniería de software juega un papel importante en el desarrollo, portabilidad, mantenibilidad, funcionalidad, fiabilidad y productividad del software, igualmente, las metodologías ágiles son procesos para desarrollar software de manera rápida con gran facilidad de adopción por los equipos de trabajo [1]. Del 11 al 13 de febrero de 2001, se llevó a cabo en Snowbird Utah una reunión de miembros prominentes de la comunidad científica donde nace el término “métodos ágiles” término que implica ser eficaz y fácil de manejar [2], 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 [3] e involucrados en el desarrollo de aplicaciones para dispositivos móviles, si es que se quiere lograr un efecto óptimo en este ámbito [4].

Sin embargo, existen muy pocas investigaciones hasta el momento sobre el uso de los métodos ágiles en el desarrollo de aplicaciones para dispositivos móviles, debido a esto, ésta investigación presenta una síntesis sobre el desarrollo de aplicaciones, sistemas operativos y metodologías de desarrollo, para lo cual, se han seleccionado las tres (3) metodologías ágiles más referenciadas [5], con mayor presencia de documentación en internet y orientadas a desarrollos de tamaño reducido propio de las aplicaciones para dispositivos móviles, como son Extreme Programming (XP), Scrum y Test Driven Development (TDD), además presenta una síntesis de metodologías usadas actualmente para dichos desarrollos. 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 así como algunas características de sus sistemas operativos, para concluir con una revisión de las metodologías utilizadas actualmente en el desarrollo de aplicaciones para dispositivos móviles.

Apreciaciones teóricas

El presente capítulo muestra una breve descripción sobre metodología del software y metodologías ágiles, enfo-

cándose en las tres (3) metodologías ágiles con mayor presencia de documentación, adaptables, con buena aceptación dentro de los equipos de desarrollo [1], [6], [7] y adaptables a las características específicas del desarrollo de aplicaciones para dispositivos móviles.

¿Qué es una metodología de desarrollo?

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.” [8].

En esta definición Avison y Fitzgerald, presentan una descripción de las metodologías de desarrollo y destacan sus principales componentes, fases, herramientas y técnicas. 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.

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” [9], 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 [10] 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 [11] y usado inicialmente para desarrollo de software, pero expandido por Boehm en 1981 [12], 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* [13] 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 Ágile Software Development, una petición por la relajación de los procesos en pro de las personas [9].

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.

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.

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 [14]. 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 colatorio [15], 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...” [14], 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 [16] [17]. Sin embargo, Conboy [3] 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.

Scrum

No se puede hablar de Scrum sin mencionar a Takeuchi y Nonaka [18] 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 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 [8].

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 [19], este equipo generalmente es de diez o menos componentes, aunque Schwaber and Beedle recomiendan equipos de cinco integrantes [19], dividiendo el equipo principal en equipos más pequeños si fuera necesario.

Aun con las recomendaciones que presentan Schwaber y Beedle [19], 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 backlog¹, 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 año 2000 Rising and Janoff publicaron tres proyectos exitosos utilizando Scrum [21].

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 [22] 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 código, cambia drásticamente la mentalidad de cualquier 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 siempre complementando a otra metodología [23] [24].

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

1. Lista de objetivos/requisitos priorizada, representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto [20]“URL”:“http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog”,“accessed”:{“date-parts”:[[“2013”,6,20]]}],“schema”:“https://github.com/citation-style-language/schema/raw/master/csl-citation.json”}.

tendencia actual es integrar TDD independientemente en cualquier metodología ya sea ágil [25] o tradicional [24] 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.

Desarrollo de aplicaciones para dispositivos móviles

Generalidades del desarrollo de aplicaciones para dispositivos móviles

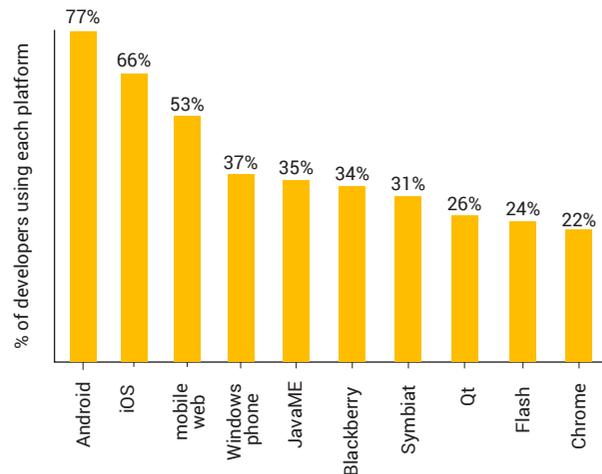
El mercado de aplicaciones móviles ha experimentado una rápida expansión durante los últimos 10 años, las plataformas móviles siguen mejorando su desempeño, y la necesidad de los usuarios de una amplia variedad de aplicaciones móviles va en aumento [26]. El desarrollo de software para las plataformas 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 [27] [28] [29] 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.

Abrahamsson [29], 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 medianas 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 [30], por tanto, deben ser desarrollados con enfoques orientados a ciclos de desarrollo relativamente cortos, propios de las metodologías ágiles para el desarrollo de software.

Sistemas operativos para dispositivos móviles

Para este apartado se han seleccionado los sistemas operativos con mayor influencia en el mercado y con un número importante de desarrolladores interesados en participar de las “App Stores”, según un estudio realizado [31] 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 1. Principales plataformas para desarrolladores Mindshare², Tomada de [31].



Android de Google

Fue adquirido por Google en 2005 cuando compro la firma Android Inc. Con el fin de asegurar que un sistema operativo para móviles (OS), pudiera ser creado y mantenido en una plataforma abierta [33], desde entonces Google invierte cada año una gran cantidad de tiempo y recursos en el proyecto Android, que ya ha demostrado ser un negocio muy beneficioso, no solo para Google, 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%

2. 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 [32].

durante el cuarto trimestre de 2012, más del doble que iOS de Apple, Inc. [34] [35]. Sino también para cualquier programador de aplicaciones para dispositivos móviles, en vista de que los usuarios de Android pueden no saber quién es el programador, pero saben lo que es Google y confían en él [33]. Debido a que su aplicación reside en el Android Market (que es controlado por Google) y/o Google Play, Google asume la responsabilidad sobre la calidad de su aplicación y por lo tanto de su distribución.

Android es un sistema operativo móvil basado en Linux que debutó formalmente en 2008 en el G1 de HTC y es actualmente desarrollado por la Open Handset Alliance, la cual es liderada por Google [35].

Una de las características especiales que posee Google, para garantizar la calidad de las aplicaciones disponibles en su Android Market, característica que también comparte con iOS de Apple, es un “kill switch” que permite el borrado remoto y global de las aplicaciones que las se consideren no aptos para sus plataformas [36]. Google ha utilizado este servicio una vez, pero lo hizo de una manera transparente y por una buena razón [37].

Mucho se podría decir sobre el creciente uso de Android, en sistemas embebidos. A pesar de que Android desplaza a Microsoft Windows y otros sistemas operativos propietarios en cuanto a número de aplicaciones que se ejecutan en teléfonos móviles y tabletas, por tanto, Google sigue siendo profundamente desinteresado frente a su competencia [38].

IOS de Apple

Sistema operativo móvil de Apple, diseñado para sus dispositivos móviles. Inicialmente desarrollado para el iPhone y presentado en público el 9 de enero de 2007, puesto en el mercado el 29 de junio de 2007, contaba con una plataforma cerrada que solo permitía instalar aplicaciones nativas desarrolladas por Apple [39] pero el 10 de julio de 2007 un grupo de hackers llamado el iPhone Dev Team lanzaron la aplicación PwnageTool, para hacer jailbreak, proceso de eliminar las limitaciones impuestas por Apple en dispositivos que utilicen el sistema operativo iOS mediante el uso de kernels modificados [40]; el IOS fue más tarde implementado para el iPod Touch, iPad y Apple TV [41]”page”:"35-40",”volume”:"7",”issue”:"1",”source”:"EBSCOhost",”archive”:"iuh",”abstract

”:"iOS is the Apple mobile operating system, for Apple mobile devices. Initially developed for iPhone, and later for iPod Touch, iPad, AppleTV. The Metropolitan University has a Web application titled eStudent, which enables students by Web to get information about their marks for all subjects, their finances, exam scheduling, professors and assistants, and send exam registration and feedback about teaching, etc. This paper explains the development of the mobile application eStudent on the iOS platform. This application enables students whenever they want, by using their iPhone mobile phone, to get access to the information from the eStudent Web application, and to present it on their iPhone User Interface (UI.

Durante la Worldwide Developers Conference (WWDC) en junio de 2012, Apple dio al mundo un primer vistazo de iOS 6, la última actualización del sistema operativo móvil de la compañía, esta versión cuenta con un montón de nuevas características y mejoras - más de 200, según Apple [42], pero ya se espera el iOS 7 preparando Jony Ive, esta versión tendrá un aspecto mucho más “plano” que iOS 6, huyendo de los iconos brillantes y acercándose más a la sobriedad y simpleza de la Metro UI de Windows Phone [43]. iOS 7 será presentada en WWDC 2013 a desarrollarse en San Francisco del 10 al 14 de junio de 2013, cumpliendo con la tradicional actualización anual del sistema [44].

En cuanto a las estadísticas, en el cuarto trimestre de 2012 el sistema operativo IOS amplió la brecha con Android tomando solo un 23.6% del mercado, frente a un 51.3% de Android [45], pero en 1er trimestre de 2013 Apple vendió 37,4 millones de teléfonos inteligentes en todo el mundo, aumentando un modesto 7% respecto al año anterior [46].

Frente al problema de la seguridad dice Barak Shrefler director de TI y gerente de seguridad en el Hospital Universitario Hadassah en Jerusalén “*Evaluamos Android y pensamos que es más vulnerable que lo iOS*” [47] especially since Android’s more open platform is being targeted by malware writers. Hardly a week goes by that security vendors hunting Android malware don’t remind us of the growing tally, as Trend Micro recently did it claiming that Android malware surged this year from 30,000 specimens in June to almost 175,000 in September. “We tested Android and we think it’s more vulnerable than

iOS,\” says Barak Shrefler, the IT and security manager at Hadassah University Hospital in Jerusalem, who said IT staff are concerned that malware or vulnerability issues around Android will simply result in future headaches, at least more than Apple iOS. At the same time, Shrefler acknowledges he’s worried about jailbroken iOS devices, too. Tamir Hardof, director of product marketing at Juniper, admitting he’s reluctant to take sides, nevertheless said “data shows there are more security threats on the Android side.” But he added that Apple’s closed system may not be what’s preferred for some enterprise customers with specific security requirements, and he’s optimistic in general that “security will improve for Android devices.””ISSN”:”08877661”,”language”:”English”,”author”:{”family”:”Messmer”,”given”:”Ellen”},”issued”:{”date-parts”:[[”2012”]]}},”schema”:”https://github.com/citation-style-language/schema/raw/master/csl-citation.json”} aunque otros expertos manifiestan su desconfianza por el jailbreak de IOS.

BlackBerry OS de BlackBerry (antes RIM)

El BlackBerry OS es un sistema operativo móvil desarrollado por Research In Motion (RIM) para sus dispositivos móviles BlackBerry. Su desarrollo se remonta a la aparición de los primeros handheld en 1999 [48], la primera BlackBerry debutó como un aparato similar a un buscador de personas en 1999 y añadió la capacidad de voz en 2002, desde entonces ha construido su reputación en base a un teclado QWERTY [49], aunque actualmente la empresa BlackBerry está apostando por una plataforma rediseñada y reinventada, disponible en dos nuevos smartphones, el BlackBerry Z10 (all-touch) y BlackBerry Q10 (táctil con teclado físico) smartphones impulsados por BlackBerry 10 que ofrecen una experiencia más rápida, más inteligente y más suave que sus antecesores [50] [51].

El 30 de enero de 2013, en la presentación del nuevo dispositivo Z10 y sistema operativo BlackBerry 10, el CEO³ de RIM, Thorsten Heins, anunció que la empresa cambiaría de nombre a BlackBerry, esta razón social entraría en vigencia el 4 de febrero de 2013 al abrir el mercado de valores [53].

3. CEO (chief executive officer), sigla utilizada para referirse al cargo ejecutivo más alto de una empresa o compañía, normalmente el Director ejecutivo [52].

Las ventas de 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, pero se espera que en 2013 el nuevo BlackBerry 10 intensifique los ingresos para la empresa [45].

Windows phone de Microsoft

Sistema operativo móvil desarrollado por Microsoft, su primera versión Windows Phone 7, fue presentada el 15 de febrero de 2010, como sucesor de la plataforma Windows Mobile, la cual estaba enfocada en un mercado de consumo empresarial y no general [54], a lo que Steve Ballmer, CEO de Microsoft, dijo: “Microsoft y sus socios están ofreciendo un tipo diferente de teléfono móvil que brinde una mejor experiencia haciendo las tareas cotidianas más rápido y con menos pasos...” [55], pero, aunque su sistema operativo fue innovador, requirió de varias actualizaciones y versiones intermedias para llegar a ser estable, aun así, no es compatible con la nueva versión 8.

Windows Phone 8, fue lanzado al mercado el 29 de octubre de 2012 como nueva apuesta de Microsoft, para competir con el iOS 7 y Android 4.3 Jelly Bean, pero, desde un punto de vista puramente técnico, las mejoras en Windows Phone 8 tendrán que recorrer un largo camino para hacer del Windows Phone OS mucho más competitivo con iOS y Android según un estudio realizado por ABI Research [56].

Samsung, HTC, Huawei y Nokia serán los principales fabricantes de teléfonos inteligentes para el Windows 8 SO, Nokia mantuvo su posición como el tercer mayor fabricante mundial de teléfonos inteligentes para el cuarto trimestre de 2012 [45], pero la cuota de mercado mundial para Windows Phone OS se ha reducido drásticamente a un 1.8% durante el año pasado [45]. La cartera de Nokia con Windows Phone 8 OS ha mejorado significativamente en los últimos meses, con nuevos modelos como el Lumia 920 que ha sido el escogido como el smartphone del año por los lectores en los premios Engadget Awards 2012, pero se cree que Nokia aún carece de un modelo que pueda ser considerado como un verdadero competidor para el iPhone de Apple o el Samsung S3 o S4 [57].

Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles

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. Hay también una cierta cantidad de 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 [58].

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 [59] 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.

Otro punto de vista [29], Abrahamsson 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.

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 [30] muy centrada en las plataformas móviles, y que se presenta con más detalle en [27]. 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 [27] 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.

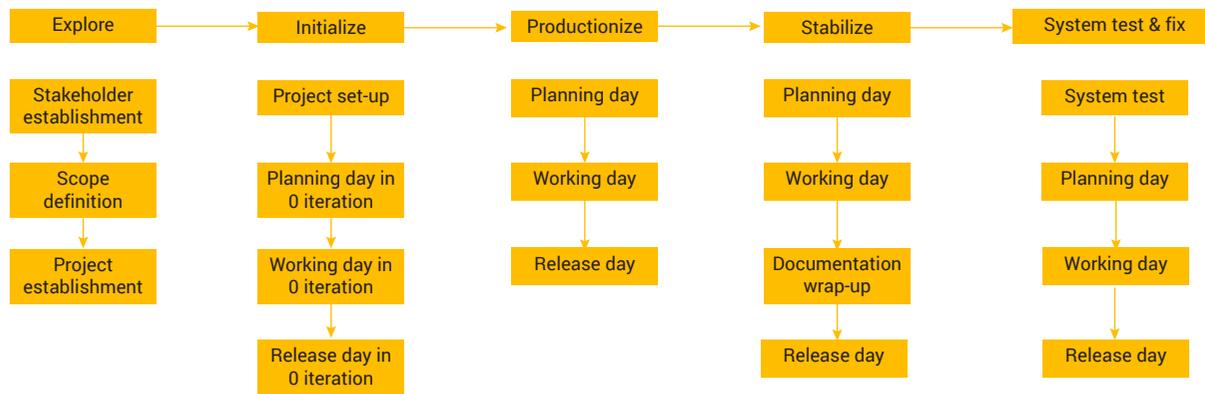
Otro punto de vista muy importante en la actualidad es la propuesta de Rahimian y Ramsin [60] HMD (Hybrid Methodology Design), la cual, se apoya en una combinación del desarrollo adaptativo de software (Adaptive Software Development, ASD) y el diseño de nuevos productos (New Product Development, NPD), 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 [61], el cual es un modelo impulsado por la usabilidad y toma como base el modelo espiral.

Mobile-D

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 [62]. 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.

Figura 2. Ciclo de desarrollo Mobile-D, Tomada de [62]



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.

Hybrid Methodology Design

Figura 3. Ciclo de desarrollo Hybrid Methodology Design, Tomada de [60]



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 [63]:

- 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 [60]. 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.

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 [61].

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.

Conclusiones

Las metodologías ágiles son una excelente alternativa para guiar proyectos de desarrollo de software de tamaño reducido, como es el caso de las aplicaciones para dispositivos móviles, gracias a la gran facilidad de adaptación que poseen; pero estas 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 de tiempo 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.

Aun cuando existe una gran variedad de metodologías (ágiles y tradicionales), en el ámbito del desarrollo para dispositivos móviles, la mayoría de los esfuerzos han sido encaminados al desarrollo de nuevas metodologías, basadas en prácticas de las tradicionales, con algunos aditamentos de técnicas modernas como usabilidad entre otras, por lo cual, esta propuesta fue orientada en las metodologías ágiles.

Referencias

- [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 Software*, vol. 2, n.o 1, p. 58, 2008.
- [2] 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.
- [3] K. Conboy y B. Fitzgerald, «Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion.», *ACM Transactions on Software Engineering & Methodology*, vol. 20, n.o 1, pp. 2:1-2:30, jun. 2010.
- [4] V. E. Jyothi y K. Nageswara Rao, «Effective Implementation of Agile Practices - Incoordination with Lean Kanban.», *International Journal on Computer Science & Engineering*, vol. 4, n.o 1, pp. 87-91, ene. 2012.
- [5] «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].
- [6] P. Abrahamsson, J. Warsta, M. T. Siponen, y J. Ronkainen, «New directions on agile methods: a comparative analysis», 2003, pp. 244-254.
- [7] «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].
- [8] D. E. Avison y G. Fitzgerald, *Information system development*. Maidenhead: McGraw-Hill Education, 2006.
- [9] «Manifiesto por el Desarrollo Ágil de Software». [En línea]. Disponible en: <http://www.agilemanifesto.org/iso/es/>. [Accedido: 13-nov-2012].
- [10] V. Szalvay, «An introduction to agile software development», *Danube Technologies*, 2004.
- [11] W. Royce, «Managing the Development of Large Software Systems: Concepts and Techniques», *WESCON Western Electronic Show and Convention*, 1970.

- [12] B. W. Boehm, *Software engineering economics*. Englewood Cliffs, N.J.: Prentice-Hall, 1981.
- [13] F. P. Brooks, *The mythical man-month : essays on software engineering*. Reading, Mass.: Addison-Wesley Pub. Co., 1995.
- [14] K. Beck y J. Zapata Martínez, *Una Explicación de la programación extrema. Aceptar el cambio*. Madrid [etc.]: Addison Wesley, 2002.
- [15] K. Beck y C. Andres, *Extreme programming explained : embrace change*. Boston, MA: Addison-Wesley, 2004.
- [16] 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.
- [17] J. Rasmusson, «Introducing xp into greenfield projects: lessons learned», *IEEE Software*, vol. 20, n.o 3, pp. 21-28, may 2003.
- [18] H. Takeuchi y I. Nonaka, «The new new product development game», *Harvard Business Review*, 1986.
- [19] K. Schwaber y M. Beedle, *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [20] «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].
- [21] L. Rising y N. S. Janoff, «The Scrum software development process for small teams», *IEEE Software*, vol. 17, n.o 4, pp. 26-32, ago. 2000.
- [22] D. Astels, *Test-driven development : a practical guide*. Upper Saddle River, N.J.; London: Prentice Hall PTR, 2003.
- [23] C. Schmidkonz y J. Staader, «Piloting of Test Driven Development in Combination with Scrum», *Scrum Alliance*, 2007. [En línea]. Disponible en: <http://members.scrumalliance.org/resources/267>. [Accedido: 18-jul-2013].
- [24] P. Letelier, J. H. Canós, y E. A. Sánchez, «An Experiment Working with RUP and XP», en *Extreme Programming and Agile Processes in Software Engineering*, vol. 2675, M. Marchesi y G. Succi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 41-46.
- [25] C. Schmidkonz, J. Staader, «Piloting of Test-driven Development in Combination with Scrum», in *Scrum Gathering Fall, 2007*. [Online]. Available: <http://www.scrumalliance.org/resources/267>. [Accessed: 13-nov-2012].
- [26] 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].
- [27] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jääliñoja, 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.
- [28] M. Satyanarayanan, «Fundamental Challenges in Mobile Computing», oct. 1998.
- [29] P. Abrahamsson, «Keynote: Mobile software development—the business opportunity of today», *Proceedings of the International Conference on Software Development*, pp. 20-23, 2005.
- [30] 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.
- [31] «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].

- [32] R. de Miguel, «Mindshare acerca el futuro del marketing móvil», *Ipmark: Información de publicidad y marketing*, n.o 787, pp. 14-16, 2012.
- [33] M. Burton y D. Felker, *Android Application Development For Dummies*. Wiley, 2012.
- [34] 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].
- [35] 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].
- [36] «Android and apps: Some favorites.», *Network World*, vol. 27, n.o 14, pp. 16-17, jul. 2010.
- [37] S. Hollister, «Google flips Android kill switch, destroys a batch of malicious apps (update)», 2011. [En línea]. Disponible en: <http://www.engadget.com/2011/03/06/google-flips-android-kill-switch-destroys-a-batch-of-malicious/>. [Accedido: 30-abr-2013].
- [38] P. Dempsey, «Carry on regardless [android operating system]», *Engineering & Technology*, vol. 6, n.o 6, p. 56, 2011.
- [39] T. Ricker, «iOS 5 jailbroken», *Engadget. Retrieved October*, vol. 26, 2011.
- [40] C. Miller, D. Blazakis, D. DaiZovi, S. Esser, V. Iozzo, y R.-P. Weinmann, *iOS hacker's handbook*. Wiley, 2012.
- [41] M. Antic, S. Jovanovic, y S. Cvetanovic, «Development of eStudent iOS Mobile Application.», *International Journal of Interactive Mobile Technologies*, vol. 7, n.o 1, pp. 35-40, ene. 2013.
- [42] S. CALDWELL, L. FRIEDMAN, y L. YAMSHON, «Get Ready for iOS 6.», *Macworld*, vol. 29, n.o 9, pp. 20-23, sep. 2012.
- [43] Mark Gurman, «Jony Ive paints a fresh, yet familiar, look for iOS 7 | 9to5Mac», 29-abr-2013. [En línea]. Disponible en: <http://9to5mac.com/2013/04/29/jony-ive-paints-a-fresh-yet-familiar-look-for-ios-7/>. [Accedido: 01-may-2013].
- [44] «WWDC - Apple Developer». [En línea]. Disponible en: <https://developer.apple.com/wwdc/>. [Accedido: 01-may-2013].
- [45] 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].
- [46] 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].
- [47] E. Messmer, «Apple iOS vs. Google Android: It comes down to security», *Network World (Online)*, p. 22, 2012.
- [48] L. Kujubu, «Handheld box provides e-mail access», *InfoWorld*, vol. 21, n.o 4, pp. 47-47, 1999.
- [49] H. McCracken, «Why the BlackBerry Is All Thumbs.», *Time*, vol. 179, n.o 5, pp. 60-60, feb. 2012.
- [50] MIGUEL ÁNGEL URIONDO, «BLACKBERRY: LA TRAVESIA DEL DESIERTO», *Actualidad Economica*, p. 56, 2012.
- [51] «Canada: BlackBerry 10: Re-designed, Re-engineered, and Re-invented», *MENA Report*, 2013.
- [52] «el CEO es el primer ejecutivo de una empresa | Fundéu BBVA». [En línea]. Disponible en: <http://www.fundeu.es/recomendacion/el-ceo-es-el-primer-ejecutivo-de-una-empresa-780/>. [Accedido: 20-jun-2013].
- [53] H. McCracken, «BlackBerry's Last Stand.», *Time*, vol. 181, n.o 5, pp. 16-16, feb. 2013.
- [54] K. Lassila, «Lisätty todellisuus Windows Phonella», *Universidad de Ciencias Aplicadas de Mikkeli*, 2012.

- [55] M. Varley, «Microsoft unveils Window Phone 7», *B & T Weekly*, 2010.
- [56] Michael Morgan, «Windows Phone 8 - Who Is It Good For? Part II | ABI Research», 25-jun-2012. [En línea]. Disponible en: <http://www.abiresearch.com/research/product/1013181-windows-phone-8-who-is-it-good-for-part-ii/>. [Accedido: 26-may-2013].
- [57] «Strategy Analytics: Global Mobile Phone Shipments Reach 1.6 Billion Units in 2012», *Business Wire*, New York, United States, New York, 2013.
- [58] O. Salo y Valtion teknillinen tutkimuskeskus, «Enabling software process improvement in agile software development teams and organisations», VTT Technical Research Centre of Finland, [Espoo, Finland], 2006.
- [59] T. Dyba y T. Dingsoyr, «What Do We Know about Agile Software Development?», *IEEE Software*, vol. 26, n.o 5, pp. 6-9, sep. 2009.
- [60] V. Rahimian y R. Ramsin, «Designing an agile methodology for mobile software development: A hybrid method engineering approach», *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*, pp. 337-342, 3.
- [61] A. Nosseir, D. Flood, R. Harrison, y O. Ibrahim, «Mobile Development Process Spiral», 2012, pp. 281-286.
- [62] «Electronics -AGILE - Agile Software Technologies», 21-nov-2012. [En línea]. Disponible en: <http://virtual.vtt.fi/virtual/agile/mobiled.html>. [Accedido: 30-abr-2013].
- [63] S. B. Kaleel y S. Harishankar, «Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges», 2013.

El Autor



Yohn Daniel Amaya Balaguera

Licenciado en Informática Educativa; Institución: Universidad pedagógica y tecnológica de Colombia; Año obtención: 2007; Magister en Tecnología Informática Institución: Universidad pedagógica y tecnológica de Colombia; Cargo: Docente, Institución: Secretaria de Educación de Boyacá; Ponente en CACIED 2013 – Congreso Andino de computación, informática y educación. San Juan de Pasto, Noviembre 5 al 8 de 2013.

Correo personal: yohn.amaya@gmail.com - yohn.amaya@uptc.edu.co