

Tipo de artículo: Artículo original
Temática: Ingeniería y Gestión de software
Recibido: 26/02/2013 | Aceptado: 12/03/2013

Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT

Comparative Evaluation of Delfdroid whit XP and Scrum using the 4-DAT

Ernesto Ávila Domenech ^{1*}, Abel Meneses Abad ²

¹ Departamento Web y Multimedia. Facultad Regional de Granma. Universidad de las Ciencias Informáticas. Ave. Camilo Cienfuegos, Manzanillo, Granma, Cuba

² Centro de Desarrollo. Facultad Regional de Granma. Universidad de las Ciencias Informáticas. Ave. Camilo Cienfuegos, Manzanillo, Granma, Cuba

{[eadomenech_sorice](mailto:eadomenech_sorice@grm.uci.cu)}@grm.uci.cu

Resumen

Debido a las diferencias notables entre el desarrollo de software tradicional y el software para dispositivos móviles, es necesario que la metodología utilizada para guiar proyectos de desarrollo de aplicaciones para móviles tenga características no tradicionales. En el presente trabajo se propone una metodología ágil de desarrollo de software específica para dispositivos móviles. Para su descripción se ha tomado la propuesta de Alistair Cockburn, en la que se indica desglosar la metodología en diez elementos como mínimo: roles, destrezas, artefactos, actividades, valores, equipos, asignación de tareas, técnicas, herramientas y estándares. Además se realiza una evaluación comparativa con las metodologías ágiles XP y Scrum mediante el método *4-Dimensional Analytical Tool* propuesto por Asif Qumer y Brian Henderson-Sellers.

Palabras clave: Desarrollo, dispositivo móvil, ingeniería de software, métodos ágiles.

Abstract

Because of the significant differences between traditional software development and software for mobile devices, it is necessary for mobile application projects to use a development methodology with non traditional characteristics. This paper proposes an agile software development methodology for mobile devices. For description is taken Alistair Cockburn's proposal, which break down the methodology described in at least ten elements: roles, skills, artifacts, activities, values, teams, assignments, techniques, tools and standards. It also shows a benchmarking between XP and Scrum using 4-Dimensional Analytical Tool proposed by Asif Qumer and Brian Henderson-Sellers.

Keywords: Agile methods, development, mobile device, software engineering.

Introducción

El desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque

el software móvil tiene que satisfacer una serie de requerimientos y condicionantes especiales que lo hace más complejo (Blanco, *et al.*, 2009).

En la literatura se pueden encontrar algunas propuestas de metodologías para el desarrollo de aplicaciones para dispositivos móviles, entre ellas se encuentran *Dynamic Channels* (Alfonso, Regateiro y Silva, 1998), *Mobile-D* (Abrahamsson, *et al.*, 2004), un modelo híbrido (RAHIMIAN; RAMSIN, 2008) y *SXP-J2ME* (Ávila, 2012). Si bien no son populares, tienen algunos elementos interesantes en comparación con las populares XP, Scrum, FDD, etc.

Específicamente *SXP-J2ME*, híbrido de las bien conocidas XP y Scrum, surgió en abril de 2012, con el objetivo de diseñar una metodología de desarrollo de software que se adecuara a las características de los proyectos que conforman la línea de productos “Aplicaciones J2ME para la Cultura y el Patrimonio” perteneciente a la Facultad Regional de la Universidad de las Ciencias Informáticas en Granma (Ávila-Domenech, 2012). La línea antes mencionada la componen los proyectos que utilizan la plataforma *Java 2 Micro Edition* (J2ME, por sus siglas en inglés) para el desarrollo de aplicaciones para dispositivos móviles y que están enfocados en la promoción y difusión de la cultura y el patrimonio de la provincia Granma.

Teniendo en cuenta que entre los dispositivos móviles más usados está el teléfono móvil y que además las aplicaciones que estos utilizan están soportadas bajo distintas plataformas, se dispuso rediseñar *SXP-J2ME* de forma tal que pudiese guiar proyectos donde se desarrollen aplicaciones para dispositivos móviles sin tener en cuenta la plataforma de desarrollo, ya sea iOS, Android, J2ME, etc; surgiendo así *Delfdroid*.

Delfdroid

En *Delfdroid* se proponen un total de ocho roles: Equipo del Proyecto (*Team*), Jefe de Proyecto (del inglés, *Project Manager*), Jefe de Producto (del inglés, *Product Management*), Arquitecto (del inglés, *Architect*), Visionario (del inglés, *Visionary*), ADD (del inglés, *Analyst-Designer-Developer*), Encargado de Pruebas (Tester) y Consultor (del inglés, *Consultant*). Además cuenta con tres fases, las mismas indican el énfasis propuesto por Barry Boehm (Boehm; y Turner, 2003) en relación a la división en dos perspectivas de desarrollo. En la Figura 1 se pueden observar las fases y flujos de trabajo propuestos.

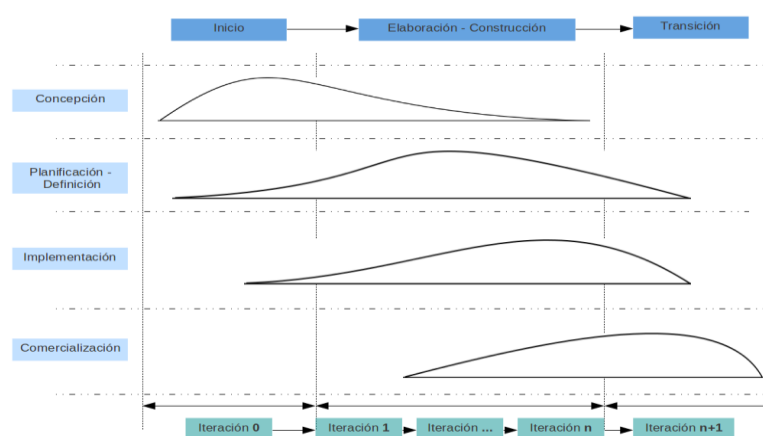


Figura 1. Fases y Flujos de Delfdroid.

Durante la fase de Inicio el Visionario deberá plasmar los objetivos y elementos que persigue la aplicación, comienza a realizar un estudio de aplicaciones similares existentes así como de los posibles usuarios. Además se realiza un diseño del escenario (diagrama o descripción que refleja la manera que se venderá o divulgará la aplicación). Una vez

terminadas las actividades anteriores se realiza un estudio de factibilidad económica de la aplicación propuesta a desarrollar teniendo en cuenta otros beneficios no necesariamente económicos.

En la siguiente fase (Elaboración-Construcción) se describen las funcionalidades a desarrollar mediante Historias de Usuario. Luego se diseñan haciendo uso de las tarjetas CRC y por último son implementadas.

Durante la fase Transición ya deberán existir varias versiones Betas en el mercado con sus respectivos beneficios; no obstante la experiencia dice que existirán inconvenientes o funcionalidades con resultados no esperados en el accionar de los usuarios por lo que habrá que realizar algunos cambios. En el caso de llegar a una versión estable y con altos niveles de aceptación se procede a registrarlo y se firma el acta de aceptación si procede.

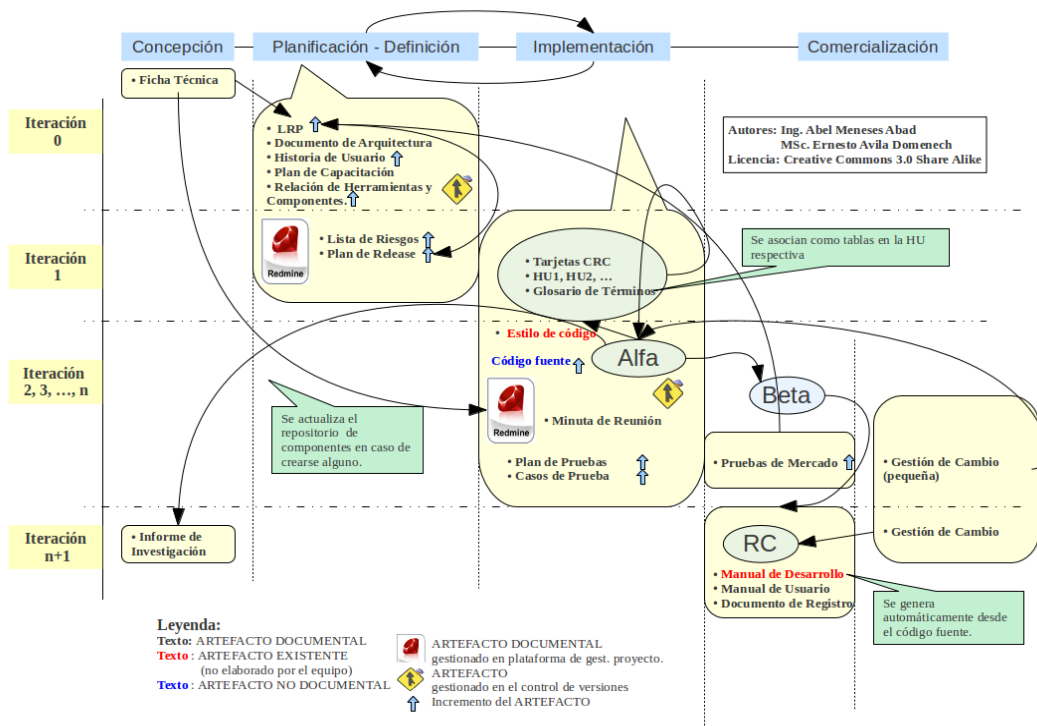


Figura 2. Flujos y Artefactos de Delfdroid.

Comparación evaluativa con XP y Scrum mediante 4-DAT

Dimensional Analytical Tool (4-DAT)

Un método de desarrollo de software se dice que es un método ágil de desarrollo de software cuando se centra en las personas, es orientado a la comunicación, flexible (listo para su adaptación a la espera de un cambio inesperado en cualquier momento), rápida (estimula el rápido e iterativo desarrollo del producto en versiones pequeñas), eficiente (se centra en acortar los plazos y costes y en la mejora de la calidad), adaptable (reacciona adecuadamente a los cambios esperados e inesperados), y aprende (centrado en la mejora durante y después del desarrollo del producto) (Qumer y Henderson, 2008b).

4-DAT facilita el examen de los métodos ágiles desde cuatro perspectivas o dimensiones: alcance de la metodología, caracterización de la agilidad, valores ágiles (Manifiesto Ágil) y caracterización del proceso de software. Aunque en la actualidad hay cuatro dimensiones evaluadas en el enfoque de 4-DAT, es ampliable en el hecho de que se pueden agregar o quitar dimensiones o elementos de las dimensiones, si se considera necesario en el futuro. (Qumer y Henderson, 2008a).

La herramienta está diseñada para comparar y analizar los métodos ágiles. Un informe que se genera con la ayuda de 4-DAT se puede utilizar para la toma de decisiones con respecto a la adopción de un método ágil apropiado. (Quer y Henderson, 2006b)



Figura 3. Dimensiones de 4-DAT.

Comparación

Para comparar Delfdroid con las metodologías de desarrollo de software XP y Scrum fueron tomados los datos de estas dos últimas metodologías de Qumer y Henderson (2006a).

Dimensión 1: Alcance de la metodología

Tabla 1. Alcance de las metodologías XP, Scrum y Delfdroid.

Criterio	XP	Scrum	Delfdroid
Tamaño de los proyectos	Pequeños y medianos	Pequeños, medianos y grandes	Pequeños y medianos
Tamaño de equipo	Menor que 10	Múltiples equipos menores que 10	Múltiples equipos menores que 10
Estilo de desarrollo	Iterativo y rápido	Iterativo y rápido	Iterativo y rápido
Estilo de código	Limpio y sencillo	No especificado	Limpio y sencillo
Entorno tecnológico	Requiere rápida retroalimentación	No especificado	Requiere rápida retroalimentación
Entorno físico	Equipos en un mismo lugar y equipos distribuidos	No especificado	Equipos en un mismo lugar y equipos distribuidos
Cultura de negocio	Colaborativo y cooperativo	No especificado	Colaborativo y cooperativo
Mecanismos de abstracción	Orientado a objeto	Orientado a objeto	Orientado a objeto

Dimensión 2: Caracterización de la agilidad

El grado de agilidad (DA) depende de los términos flexibilidad (FY), velocidad (SD), eficiencia (LS), aprendizaje (LG) y adaptabilidad (RS).

Para calcular la agilidad en esta dimensión Qumer y Henderson-Sellers proponen la siguiente fórmula: (Qumer y Henderson, 2008b).

$$DA(\text{Object}) = (1/m) \sum m DA(\text{del ingles, Object, Phase or Practices})$$

Tabla 2. Grado de Agilidad de Delfdroid.

		Características de agilidad					
Delfdroid		FY	SD	LS	LG	RS	Total
(i) Fases							
1	Inicio	1	1	0	0	1	3
2	Elaboración-Construcción	1	1	0	1	1	4
3	Transición	1	1	0	1	1	4
Total		3	3	0	2	3	11
Grado de Agilidad		3/3	3/3	0/3	2/3	3/3	11/(3*5)
(ii) Prácticas							
1	Desarrollo iterativo e incremental mediante Sprint cortos	1	1	1	1	1	5
2	Programación en pares	1	0	0	1	1	3
3	Pruebas	1	1	0	1	1	4
4	40 horas semanales	0	0	0	1	0	1
5	Utilización de Metáforas	0	1	1	0	0	2
6	Rápida retroalimentación	1	1	1	1	1	5
7	Diseño simple	1	1	1	1	1	5
8	Refactorización	1	1	1	1	1	5
9	Participación activa de todos los miembros del proyecto	1	1	1	1	1	5
10	Creación de varios modelos en paralelo	1	1	0	1	1	4
11	Reutilización constante	0	1	1	0	1	3
12	Estándar de codificación	1	1	1	1	1	5
13	Reunión para controlar el Sprint terminado y planificar el siguiente	1	1	0	1	1	4
14	Reportando progreso	1	1	1	1	1	5
15	Cliente en la puerta	1	0	1	1	1	4
Total		12	12	10	13	13	60
Grado de Agilidad		12/15	12/15	10/15	13/15	13/15	60/(15*5)

Delfdroid al ser un híbrido de las metodologías ágiles XP y Scrum, posee prácticas que estas dos últimas también poseen, o al menos una de ellas. Las enumeradas con 1, 2, 3, 4, 5, 7, 8 y 12 han sido tomadas de XP por lo que se han evaluado de igual manera que en Qumer y Henderson (2008a). De forma semejante la enumerada con 13 ha sido evaluada, con la diferencia de que pertenece a Scrum.

Teniendo en cuenta que al realizar reuniones para controlar el Sprint terminado y planificar el siguiente se realiza un juego de planificación, se puede decir que la enumera con 1 pertenece tanto a XP como a Scrum. Por otra parte se consideran a las prácticas 6, 9, 10 y 15 como nativas de Delfdroid.

Dando oportunidad a todo el equipo que participe activamente podrán proponer funcionalidades nuevas al sistema o sencillamente cambios significativos que el Visionario no ha sugerido. Esta acción es flexibilidad, veloz, eficaz, el equipo aprende con el aumento de la experiencia y perfectamente posee adaptabilidad por lo que ha evaluado de forma global con 5.

Al sacar un producto funcional aunque con un número importante de requisitos por cumplir se poseen criterios de usuarios finales de manera rápida; pudiendo el equipo retroalimentarse a una mayor velocidad. Teniendo en cuenta las características a evaluar, se le otorga de manera general la evaluación de 5.

Por último, a diferencia de XP, no se considera al cliente en el equipo por la sencilla razón de que no existe tal cliente. Delfdroid está enfocado a los usuarios. La práctica cliente en la puerta es una metáfora que indica que no existe

cliente dentro ni fuera del equipo del proyecto. Como se ha explicado, se toman en cortos periodos de tiempo, a usuarios finales y se les solicita que comenten, critiquen y propongas ya sea cambios o nuevas funcionalidades. A diferencia de las explicadas anteriormente, se considera que esta práctica no le da velocidad por lo que su evaluación global es 4.

Dado el grado de agilidad de XP y Scrum se obtienen los siguientes resultados.

Tabla 3. Comparación del grado de agilidad entre XP, Scrum y Delfdroid.

	XP	Scrum	Delfdroid
Fases	0.70	0.60	0.73
Posición	2	3	1
Prácticas	0.73	0.80	0.80
Posición	3	1	1
Promedio (Fases y Prácticas)	0.72	0.70	0.77
Posición	2	3	1

Dimensión 3: Valores ágiles

La tercera dimensión de 4-DAT es sólo cualitativa y presentan los valores ágiles que son promulgadas por la metodología ágil bajo investigación. Se puede observar que tanto XP como Scrum ofrecen soporte para todos los valores iniciales ágiles, pero no son compatibles con los dos identificados recientemente por Qumer y Henderson Sellers. XP no ofrece ningún apoyo, ya sea para "mantener el proceso ágil" (lo que podría sugerir un eslabón perdido de la SPI (mejora de procesos software) de la comunidad) o para "mantener el coste del proceso efectivo" (un valor pragmático para la ágil adoptabilidad comercial). Esta última característica no se ve bien en Scrum. (Qumer y Henderson, 2008a).

En el caso de Delfdroid, específicamente en el manteniendo del costo efectivo del proceso, se observa se le da gran importancia a la retroalimentación. Mediante ella se conoce si la aplicación tiene una buena aceptación por parte de los usuarios y en correspondencia una ganancia económica para la empresa desarrolladora de dicha aplicación. Otro elemento es el desarrollo de varios modelos en paralelo; contribuyendo así a que llegue a los usuarios el producto que realmente quieran tener sin una demora considerable. Una de las características que ayuda a esta rapidez es la reutilización.

Tabla 4. Grado de agilidad de XP, Scrum y Delfdroid en la dimensión 3.

Valores Ágiles	XP	Scrum	Delfdroid
Individuos e iteraciones por encima de procesos y herramientas	-Juego de planificación. -Propiedad colectiva. -Cliente en el equipo de desarrollo. -Programación en pares.	-Scrum teams. -Sprint planning meeting. -Daily scrum meeting.	-Participación activa de todos los miembros del proyecto. -Propiedad colectiva. -Reunión para controlar el Sprint terminado y para planificar el siguiente. -Cliente en la puerta.
Software activo encima de documentación comprensiva.	-Release cortos. -Pruebas. -Integración continua.	-Sprint. -Revisión de Sprint.	-Sprint. -Entregas pequeñas mediante Sprints cortos. -Pruebas -Reunión para controlar el Sprint terminado y para planificar el siguiente. -Rápida retroalimentación. -Refactorización.
La colaboración con el cliente más que la negociación de un contrato.	-Juego de planificación. -Integración continua.	-Product backlog. -Sprint planning meeting.	-Desarrollo iterativo e incremental. -Participación activa de todos los miembros del proyecto.
Responder a los cambios más que seguir estrictamente un plan	-Metáforas. -Simple diseño. -Refactorización. -Estándar de codificación.	-Sprint review. -Sprint planning meeting.	-Utilización de Metáforas. -Simple diseño. -Refactorización. -Rápida retroalimentación. -Creación de varios modelos en paralelo. -Estándar de programación. -Reunión para controlar el Sprint terminado y planificar el siguiente.
Manteniendo procesos ágiles.		-Sprint review. -Daily scrum meeting.	-Reunión para controlar el Sprint terminado y planificar el siguiente. -Participación activa de todos los miembros del proyecto. -Rápida retroalimentación.
Manteniendo del costo efectivo del proceso.			-Rápida retroalimentación. -Creación de varios modelos en paralelo. -Reutilización constante.

Dimensión 4: Caracterización del proceso de software

En esta parte de la evaluación, se examinan las prácticas de XP, Scrum y Delfdroid para el apoyo a los procesos de software. En la Tabla 5 se presenta un informe de evaluación para esta cuarta dimensión. Al igual que en la Dimensión 3, la comparación es puramente cualitativa e informativa (descriptiva y no normativa).

Tabla 5. Proceso de software de XP, Scrum y Delfdroid. (Dimensión 4).

Proceso de software	XP	Scrum	Delfdroid
Proceso de desarrollo	-Release cortos. -Metáforas. -Diseño simple. -Pruebas. -Refactorización. -Programación en pares. -Propiedad colectiva. -Integración continua. -Cliente en el equipo de desarrollo. -Estándar de codificación.	-Scrum teams. -Product backlog. -Sprint. -Sprint review.	-Entregas pequeñas mediante Sprints cortos. -Utilización de Metáforas. -Diseño simple. -Pruebas. -Refactorización. -Programación en pares. -Propiedad colectiva. -Desarrollo iterativo e incremental. -Cliente en la puerta. -Estándar de codificación. -Rápida retroalimentación.
Proceso de gestión de proyecto	-Juego de planificación.	-Scrum Master. -Sprint planning meeting. -Daily scrum meeting.	-Participación activa de todos los miembros del proyecto. -Reunión para controlar el Sprint terminado y planificar el siguiente.

Proceso de la configuración del software	No especificado	No especificado	No especificado
Proceso de gestión de procesos	No especificado	No especificado	No especificado

Conclusiones

Luego de describir Deldroid se puede concluir que se ha propuesto una metodología con tres fases y cuatro flujos de trabajo principales. Por sus características se considera ágil y adaptable a entornos de desarrollo de aplicaciones para dispositivos móviles. Además agrupa muchas de las llamadas “buenas prácticas de desarrollo de software” empleadas en las más importantes y utilizadas metodologías ágiles encontradas en la literatura.

Referencias

- ABRAHAMSSON, P.; HANHINEVA, A.; HULKKO, H.; *et al.* Mobile-D: an Agile Approach for Mobile Application Development. In *Companion to the 19th annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications* ISBN 1-58113-833-4. [En línea]. New York, NY, USA: ACM, 2004. Disponible en: [\[http://doi.acm.org/10.1145/1028664.1028736\]](http://doi.acm.org/10.1145/1028664.1028736).
- ALFONSO, A. P.; REGATEIRO, F.S. y SILVA, M.J. Dynamic Channels: a New Development Methodology for Mobile Computing Applications. 1998.
- ÁVILA-DOMENECH, E. Propuesta de metodología de desarrollo de software para su utilización en la línea de productos “Aplicaciones J2ME para la Cultura y el Patrimonio”. Camagüey: Universidad de Camagüey, 2012. 104.
- BLANCO, P.; CAMARERO, J.; FUMERO, A.; *et al.* Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone. Universidad Politécnica de Madrid, 2009.
- BOEHM, B y TURNER, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison Wesley, 2003. 304. ISBN 0-321-18612-5.
- QUMER, A; HENDERSON-SELLERS, B. A Framework to Support the Evaluation, Adoption and Improvement of Agile Methods in Practice. *Elsevier Inc.* 2008a, s. 1899–1919. Dostupné také z: Disponible en: [\[www.sciencedirect.com\]](http://www.sciencedirect.com).
- QUMER, A. y HENDERSON-SELLERS, B. An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering. *Elsevier Inc.* 2008b, s. 280–299.
- QUMER, A y HENDERSON-SELLERS, B. Comparative Evaluation of XP and SCRUM using the 4D Analytical Tool (4-DAT). Alicante, Spain, 2006a.
- QUMER, A; HENDERSON-SELLERS, B. Measuring Agility and Adoptability of Agile Methods: A 4-Dimensional Analytical Tool. 2006b. ISBN 972-8924-09-7.
- RAHIMIAN, V y RAMSIN, R. Designing and Agile Methodology for Mobile Software Development: a Hybrid Method Engineering Approach. Marrakech, 2008.