

Desarrollo de un App educativa con ludificación aplicando metodologías ágiles

Fernández Tessier, Carlos; Ramiro Córdoba, Ernesto

Ciclos Formativos de Grado Superior del Departamento de Informática y Comunicaciones
Escuela Politécnica
Universidad Europea
c/ Tajo s/n Urb. El Bosque 28670-Villaviciosa de Odón- (Madrid)
e-mail: carlos.fernandez@uem.es; ernesto.ramiro@uem.es,
web: <http://www.uem.es>

Resumen. *En el presente proyecto integrador para el curso 2014-15 se propone al alumnado desarrollar una aplicación para dispositivos móviles. Se trata de una herramienta educativa bilingüe dirigida al campo editorial, como complemento a sus libros de Formación Profesional. Está pensada para móviles y tabletas Android e incluirá elementos de ludificación. Este proyecto integrador a su vez se enmarca en los módulos de “Acceso a Datos”, “Desarrollo de Interfaces” y “Programación Multimedia y Dispositivos móviles” de 2º curso de la titulación “Desarrollo de Aplicaciones Multiplataforma”. Se trabaja además en ella con el framework de metodología ágil Scrum que utilizan las empresas de desarrollo de software para gestionar el desarrollo de productos. Una vez asignados roles, los equipos se organizarán por sí mismos fomentándose la colaboración de todos los miembros, así como una comunicación diaria cara a cara. La clave de las metodologías ágiles es la adaptación al cambio y la búsqueda de soluciones a los nuevos problemas que puedan surgir según las necesidades del cliente, incorporando los cambios imprevistos no especificados en la planificación original.*

Palabras clave: Ciclos Formativos de Grado Superior, ASIR, Aprendizaje basado en Proyectos, Proyecto integrador, competencias transversales, PBEs, SCRUM, Android.

1. INTRODUCCIÓN

Este artículo está inspirado en la categoría de Metodologías de aprendizaje propuesta en la XII Jornada Internacional de Innovación Educativa de la Universidad Europea.

Actualmente, desde la UEM se está impulsando una escuela politécnica basada en proyectos integradores ([ver más sobre PBL, Project Based Learning](#)), con el objetivo de preparar a los alumnos para el mundo laboral, donde deberán enfrentarse a proyectos de

software en los que será fundamental la colaboración en el equipo de desarrollo y la adaptación a los cambios solicitados por el cliente.

En este caso, el proyecto nace de la necesidad real de una editorial de Formación Profesional, interesada en que los alumnos implementaran una aplicación bilingüe para dispositivos móviles y tabletas incluyendo en ella elementos de ludificación.

Para conseguir la colaboración de los alumnos y la adaptación a los cambios se ha utilizado el framework de metodología ágil *Scrum* (JANOFF y RISING 2000), utilizado en empresas de desarrollo de software para gestionar el desarrollo de productos. La clave en las metodologías ágiles es la adaptación al cambio y la capacidad para encontrar soluciones a los nuevos problemas que puedan surgir según las necesidades del cliente, considerando los cambios imprevistos que pueden no haber sido especificados en la planificación original.

En las empresas de software la mayoría de actividades se realizan en grupo; sin embargo, en la mayoría de centros con ciclos de Desarrollo de Aplicaciones Multiplataforma se les pide a los alumnos que demuestren individualmente mediante un examen si son aptos, lo que genera una gran competitividad y falta de solidaridad. Por esta razón se decidió formar equipos equilibrados con la responsabilidad de organizarse por sí mismos, fomentándose así el aprendizaje cooperativo entre los integrantes de los equipos y colaborativo entre los grupos (PANITZ,1997). En este contexto se facilita el llamado Peer Learning (ver más sobre Peer Learning) o aprendizaje entre iguales, que es una oportunidad para unos y otros: el alumnado se convierte en generador de conocimientos, fortaleciendo tanto sus competencias técnicas como otras competencias transversales; de forma natural se intercambia información con ánimo de aprender y se plantean dudas e inquietudes entre alumnos y profesor de forma horizontal, de manera que la comunicación y los conocimientos adquiridos son favorecidos al utilizar un lenguaje y unas inquietudes comunes.

1. OBJETIVOS

Con la idea de trabajar las competencias que se persiguen en las programaciones didácticas de los Módulos señalados, nos marcamos una serie de objetivos como resultados del aprendizaje.

1.1. Objetivos generales

Crear una aplicación para que funcione en dispositivos móviles de diferentes tamaños (tanto teléfonos como tablets) con pantalla táctil basada en Android.

1.2. Objetivos específicos

- Gestionar y realizar el trabajo del área asignada, en relación con las cargas de trabajo propias del Ciclo Formativo (Planificación del tiempo).
- Realizar consultas a la persona adecuada y saber respetar la autonomía de los compañeros.
- Mantener el espíritu de innovación y actualización en el ámbito de su trabajo, buscando información para la resolución de problemas de forma autónoma y mostrando empatía, habilidades para la sinergia de equipos y liderazgo.
- Gestionar de forma eficaz los conflictos que se puedan producir, mediando y contribuyendo al establecimiento de un ambiente de trabajo agradable y actuando en todo momento de forma sincera, respetuosa y tolerante.
- Resolver problemas y tomar decisiones individuales, siguiendo las normas y procedimientos establecidos, definidos dentro del ámbito de su competencia.

2. METODOLOGIA

Como se explicó anteriormente, para el proyecto se utilizó la metodología *Scrum*, un framework de desarrollo ágil de software para gestionar el desarrollo de productos.

Scrum se define como una estrategia flexible e integral de desarrollo del producto donde un equipo trabaja como unidad para alcanzar un objetivo, los retos asumen un enfoque secuencial del desarrollo de producto; se habilita a los equipos para organizarse por sí mismos animando a la colaboración presencial u online de todos los miembros del equipo, así como a una comunicación diaria cara a cara de todos los miembros y perfiles en el proyecto.

Una idea clave del *Scrum* es el reconocimiento de que, durante el desarrollo del proyecto, el cliente puede cambiar de idea sobre lo que quiere o necesita y que puede haber cambios imprevistos no especificados en la planificación original. *Scrum* adopta una aproximación práctica, que plantea que el problema no se puede entender de forma integral desde un principio. La adaptación al cambio y el poder encontrar soluciones a los nuevos problemas (SAVERY y DUFFY, 2001) supone un desafío y una oportunidad para la capacidad de los miembros del equipo de desarrollo.

También se utilizaron algunos principios de la programación extrema:

- Desarrollo iterativo e incremental.
- Desarrollo guiado por pruebas (TDD) y pruebas unitarias.
- Programación en parejas (Pair Programming).
- Integración del equipo de programación con el cliente.
- Refactorización del código.
- Simplicidad en el código.

- Propiedad del código compartida en un repositorio (Github).

En el aprendizaje cooperativo se fomenta ante todo la participación. Primero el educando se enfrenta al problema solo, a través de una participación proactiva según la organización de situaciones de aprendizaje de Jean Piaget. A continuación, para aprender significativamente necesitará de la ayuda de otros; según Lev S. Vygotsky esto se consigue mediante el aprendizaje cooperativo. El aprendizaje cooperativo necesita estos dos momentos, que deben ser alternados didácticamente, fomentando tanto el trabajo individual como en equipo.

Ya desde el punto de vista de las metodologías ágiles los profesores tenían el papel de *Product Owner* y *Scrum Master*; éste último se encarga de proponer tareas e indicar a cada alumno/a el rol que ejercerá dentro del grupo -siendo estos roles rotativos en cada Sprint- responsabilizándose cada uno de la solución de una parte del problema (tarea o User Story). De esta forma, los profesores son quienes en todo momento diseñan y mantienen la mayor parte de la estructura de las interacciones y de los resultados que se han de presentar pública y regularmente en los *Sprint reviews* (véase Fig. 1)

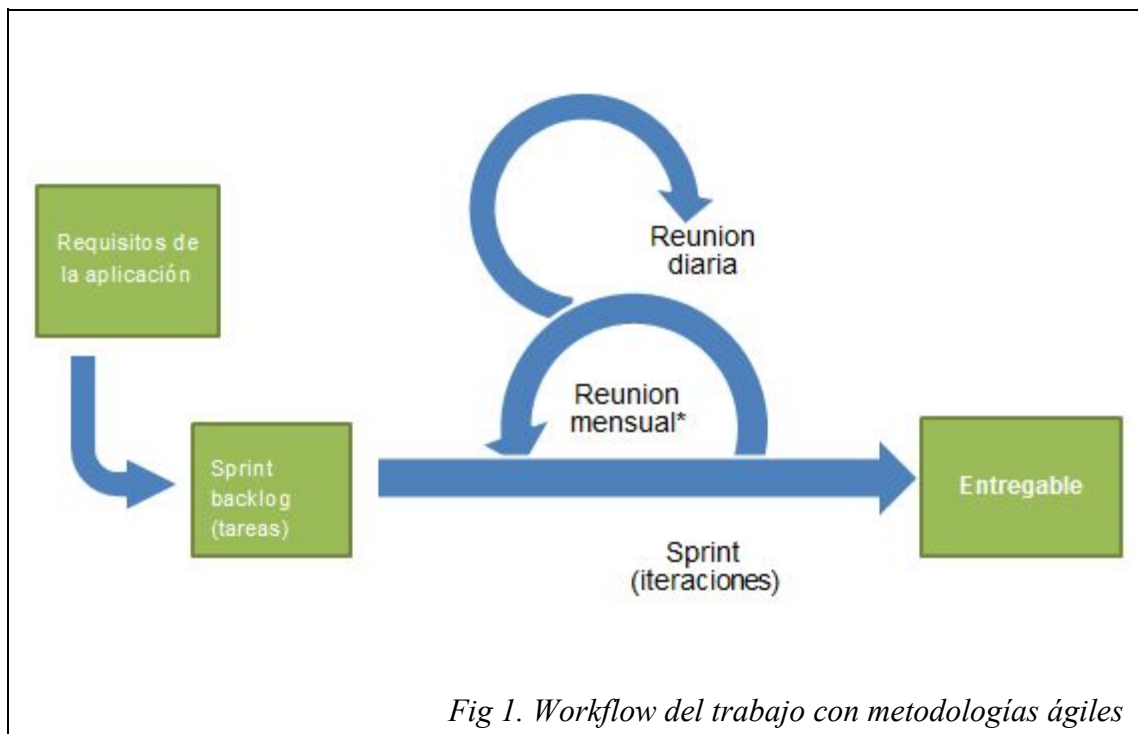


Fig 1. Workflow del trabajo con metodologías ágiles

Además se fomentó también el aprendizaje colaborativo entre equipos, en el que los grupos comparten entre ellos sus avances y dificultades (DILLENBOURG, 1999). Por otro lado, cada miembro es responsable tanto de su tarea/ individual como de observar las de los demás miembros del grupo. Para ello, los profesores pedimos que al finalizar cada *Sprint* plasmaran una nota individual y otra grupal cuantitativa y cualitativa

argumentadas.

Este tipo de aprendizaje se enmarca con la filosofía del software libre; de forma coherente, desde el principio del desarrollo los alumnos han subido el código de sus proyectos a Github ([ver más sobre GitHub](#)) con una licencia GPL.

3. DESARROLLO

3.1 Temporalización

El proyecto se desarrolló a lo largo del primer y segundo trimestre y comenzó a finales de Octubre con una investigación sobre tecnologías a utilizar, desarrollo de las primeras maquetas (*Mockups*) y planificación del *Backlog*. Para el segundo trimestre se inició el desarrollo práctico de la aplicación sobre Android Studio.

A mediados de Diciembre, -al final del primer trimestre- se realizó una breve presentación de estudio de casos de aplicaciones educativas con ludificación.

El desarrollo del proyecto se definió sobre la idea de *Sprints*. Al final de cada *Sprint* (cuatro *Sprints* en total, uno cada 2 ó 3 semanas) se realizó una exposición de los objetivos conseguidos (*Sprint Retrospective*), a la que estuvo invitado el alumnado de primero de DAM, en la cual pudieron intercambiar opiniones con los equipos sobre el desarrollo de aplicación.

Previamente -al comienzo de las cuatro fases- se crearon los equipos de cuatro miembros y se generó un acta de constitución que recogía a los miembros que componían cada equipo.

Algunas tareas crearon dependencias sobre el inicio de otras. Estas se especificaron, para que tanto los equipos como los profesores responsables de cada módulo en su rol de Scrum Master tuvieran especial cuidado en el cumplimiento de los plazos, con el objeto de no bloquear el desarrollo del proyecto.

En cada equipo y durante cada *Sprint*, de forma rotativa, se eligió un responsable de la gestión del repositorio y del *Backlog*; un responsable de realizar las pruebas de los diferentes elementos y de su validación, y un tercer responsable de ir documentando la aplicación.

Al trabajar sobre un repositorio online abierto como GitHub, los grupos pudieron

mantener junto al código un blog público con el avance del desarrollo, como en el siguiente ejemplo:



Fig 2. Vista de la wiki del proyecto en GitHub (click to open)

Al final de cada *Sprint* se realizó una presentación de los proyectos, apoyándose en el blog de cada uno (véase Fig. 2) en el que se analizaban los objetivos conseguidos. En esta presentación se identificaban las oportunidades de mejora y se compartían los logros y la línea de trabajo con el resto de equipos (DT o *Development Teams*).

3.2. Entrega

Los cuatro equipos presentaron en el último Sprint de finales de Marzo su aplicación; todos lograron cumplir los requisitos mínimos pedidos:

- Un aplicación funcional e instalable según las especificaciones de cada asignatura.
- Una página wiki en el repositorio Github.
- Presentación y defensa de la app.

Todas las apps, aunque se encontraban en una fase alpha, eran completamente funcionales; después de la presentación se hizo una demo en un dispositivo android virtualizado para facilitar la tarea de mostrar las funcionalidades y características de la aplicación en un proyector de vídeo. En algunos casos también se realizaron instalaciones en dispositivos físicos (móviles y tablets).

Algunos equipos sobrepasaron las expectativas con un diseño muy profesional y funcionalidades brillantes. La mayor parte de los proyectos se pueden descargar en los móviles y tabletas desde [Google App Store](#).

4. CONCLUSIONES

Durante el desarrollo del proyecto, tanto profesorado como alumnos se han encontrado con algunas dificultades. Por parte de los profesores se han encontrado los siguientes problemas:

- Dificultad en adaptar el currículo de los módulos a las características del proyecto; no se ha podido adaptar el 100% de los contenidos.
- Nivel técnico muy alto, en el que se ha hecho un esfuerzo importante en utilizar las últimas herramientas y novedades tecnológicas.
- Gran carga de trabajo.

Por otro lado los alumnos también han encontrado dificultades:

- Falta de experiencia en la metodología Scrum y en el trabajo colaborativo.
- Problemas al trabajar con los repositorios en Github.
- Dificultad para alcanzar los objetivos de los sprints por falta de tiempo.
- Conflictos surgidos entre algunos miembros del equipo que faltaban a clase y/o no colaboraban con el grupo.

A pesar de todo, el balance ha sido positivo, obteniéndose los siguientes logros por parte del profesorado:

- Mejora de la coordinación entre los profesores.
- Aprendizaje horizontal: profesores y alumnos aprenden a la vez, asumiendo los profesores el rol de guía.
- Las reuniones continuas para evaluar el proyecto y los conflictos surgidos en los grupos han mejorado el acercamiento con los alumnos.

En el caso de los alumnos, los principales logros han sido los siguientes:

- La mayoría de los alumnos se han implicado en el proyecto, incluyendo a los alumnos con más dificultades.

- Todos ellos han realizado un esfuerzo muy importante, trabajando tanto en clase como fuera del aula.
- Han mejorado en el trabajo cooperativo y colaborativo y aprendido a resolver los conflictos internos, y han ganado en madurez para afrontar problemas futuros en el trabajo.
- Se han trabajado las competencias transversales:
 - Comunicativas: Presentación del proyecto en los Sprints Reviews.
 - Delegación: Reparto de tareas.
 - Tutoría entre iguales: Explicar y ayudar a los compañeros con más dificultades.
 - Curiosidad: Buscar las últimas tecnologías y aplicar conceptos no vistos en clase.
 - Escritura: Escribir en la página wiki la evolución del proyecto.

En cuanto a las metodologías ágiles y a Scrum en particular, detectamos la necesidad de una mayor flexibilidad a la hora de definir las tareas de cada Sprint (que al principio de cada etapa se vuelcan en una pizarra de nombre *Kanban*).

No es fácil dimensionar el tiempo que requiere una tarea ni las dependencias que pueden aparecer con otras tareas, ni siquiera asignar una prioridad o descubrir que hay que eliminar características deseables para alcanzar los plazos. En ocasiones también se hace necesario incorporar nuevas funcionalidades no previstas. En ese sentido es tarea del Scrum Master, en contacto con el Product Owner, rediseñar la planificación y adaptarla a lo posible aunque no sea lo deseable.

En la imagen vemos las tareas en el *Kanban* (véase Fig. 3) en este caso se definían al inicio del proyecto, se asignaban a cada Sprint y se hacían avanzar por los estados (backlog > ready > progress > done). Desde su definición inicial en el backlog hasta su incorporación a los Sprints, y durante el desarrollo de los mismos, las tareas no podían modificarse.

Vemos sin embargo la necesidad de flexibilizar la incorporación de entrada/salida de tareas al *Kanban* siguiendo los parámetro de la metodología Scrum Ban ([ver más sobre Scrum Ban](#))



Fig 3. Tablero Kanban con las fases y evolución del desarrollo

REFERENCIAS

DILLENBOURG, P. Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series. Elsevier Science, Inc., 1999.

N.S. JANOFF; L. RISING(2000). "The Scrum Software Development Process for Small Teams" (PDF). Retrieved May 16, 2015.

PANITZ, T. Collaborative versus cooperative learning: Comparing the two definitions helps understand the nature of interactive learning. cooperative learning and college teaching, 1997. URL <http://home.capecod.net/~tpanitz/tedsarticles/>

Qué es Scrum. *Proyectos ágiles*. URL <http://www.proyectosagiles.org/que-es-scrum>. Retrieved May 16, 2015,

SAVERY, J. R. y DUFFY, T. M. Problem based learning: An instructional model and its constructivist framework, 2001. URL <http://cee.indiana.edu/publications/journals/TR16-01.pdf>. Retrieved May 16, 2015.