

Un Primer Paso a la Agilidad: Retrospectivas para el Aprendizaje de la Ingeniería del SW

Jennifer Pérez, Agustín Yagüe, Jessica Díaz, Santiago Alonso

Departamento de Organización y Estructura de la Información
E.U. Informática. Universidad Politécnica de Madrid (UPM)
Madrid, España

jenifer.perez@eui.upm.es, agustin.yague@upm.es,
yesica.diaz@upm.es, salonso@eui.upm.es

Resumen. En los últimos años la industria software demanda, cada vez más, ingenieros que posean conocimientos y experiencia en la aplicación de metodologías ágiles. Los principios y valores en los que se basan las metodologías ágiles fomentan la adquisición de competencias como la capacidad de organización, el trabajo en equipo, la comunicación, o el liderazgo, entre otras, denominadas en el marco del Espacio Europeo de Educación Superior (EEES) como competencias generales o transversales. Ambas razones justifican la adopción de las metodologías ágiles como métodos de aprendizaje activos, es decir, la implantación de metodologías ágiles durante el ciclo formativo del ingeniero software. Esta apuesta se ha materializado en el proyecto de innovación educativa *Agile Learning* - Aprendizaje Ágil, cuyos primeros pasos, resultados y lecciones aprendidas se presentan en este artículo.

1 Introducción

El dinamismo y variabilidad de la actual industria software requiere la adaptación de las formas de desarrollar aplicaciones para ajustarse a un mercado cambiante y a la reducción de la vida de los productos [1]. El ciclo de vida de los productos y servicios software se acorta, lo que obliga a incrementar la productividad, a disminuir el tiempo de reacción, y a adaptarse rápidamente a los cambios y a las nuevas necesidades de los clientes. Los métodos ágiles surgieron a partir del *Manifiesto Agile* [2] como solución a las necesidades de la industria software actual. De hecho, desde el punto de vista de las metodologías ágiles el cambio se percibe como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente; la gestión del cambio se convierte en un aspecto inherente al propio proceso de desarrollo software, mejorando así su adaptación a entornos cambiantes.

En la última década, el desarrollo ágil de software ha adquirido una gran relevancia en el área de la industria software, debido a que su adopción en las empresas está revirtiendo en un aumento de la calidad y competitividad en el mercado [3]. En los últimos años, este evidente interés a nivel internacional por la adopción de metodologías ágiles [3], se está trasladando al ámbito nacional. Pero para dar un paso adelante y llegar a equipararnos nacionalmente en esta adopción ágil con otros países, es necesario que nuestros ingenieros de software no sólo dispongan de habilidades y

conocimientos sobre la implantación de un desarrollo convencional, sino también del desarrollo ágil. Es por ello, que la primera semilla o primer paso debemos darlo en su formación en ingeniería del software, de manera que los ingenieros de software que se incorporen al mundo laboral tengan las habilidades y conocimientos necesarios para poner en marcha metodologías ágiles. Esto hace indispensable que los contenidos necesarios para el aprendizaje de las metodologías ágiles se impartan en asignaturas pertenecientes a la rama de ingeniería del software. Pero además, en este artículo se presenta una apuesta hacia la agilidad todavía mayor, la adopción de metodologías durante todo el ciclo formativo del ingeniero de software. Esta apuesta se ha materializado en un proyecto de innovación educativa llamado *Agile Learning - Aprendizaje Ágil* [4]. El fundamento de este proyecto y el marco perfecto para su aplicación es la necesidad actual de la universidad española de adaptarse al Espacio Europeo de Educación Superior (EEES).

El EEES marca unas directrices a seguir, que afectan, no solo a la forma en que el alumno debe aprender, sino que también afectan a la forma en que el profesor debe enseñar, caracterizándose principalmente por un modelo de evaluación continua y modelo de aprendizaje basado en competencias, tanto generales como específicas.

Hasta ahora, la universidad se había ocupado únicamente en las competencias específicas, es decir, la adquisición y desarrollo de conocimientos por parte del alumno. Sin embargo, el gran reto y cambio de la adaptación al EEES radica en la adquisición por parte del alumnado de competencias generales, es decir, capacidades, habilidades y/o aptitudes que el alumno debe desarrollar para aplicarlas a lo largo de su carrera profesional. Para cumplir dichas competencias generales se necesitan de métodos de aprendizaje activos que permitan que los alumnos desarrollen capacidades como: (i) Capacidad de organización y planificación; (ii) Comunicación; (iii) Liderazgo, (iv) Capacidad de evaluación y auto-evaluación; (v) Trabajo en equipo; (vi) Entrega de trabajos de calidad para el profesorado (vii) Seguimiento del proceso de aprendizaje del alumno, etc.

En este sentido, las metodologías ágiles proporcionan unos principios, valores, y prácticas que se presentan como una solución para esta necesidad [5]. Esto es debido a que por un lado, posibilitan la adquisición de todas estas competencias de forma flexible y sencilla, y por otro lado, permiten que los ingenieros de software apliquen una metodología ágil durante toda su formación. Esto permitirá no sólo mejorar la preparación de nuestros ingenieros de software, sino que también, cuando un ingeniero de software se incorpore al mundo laboral, tendrá la experiencia de 4 años de aplicación de metodologías ágiles.

En este artículo presentamos la idea global del proyecto *Agile Learning - Aprendizaje Ágil*¹, el cual ha adoptado la metodología ágil Scrum [6, 7] y la ha adaptado a un marco de trabajo académico. En concreto, en este trabajo se presenta la implantación del trabajo en equipo, de entregas frecuentes en iteraciones incrementales y de reuniones retrospectivas. El proyecto ha sido aplicado en algunas asignaturas impartidas en el Departamento de Organización y Estructura de la Información (OEI) [8] de la Escuela Universitaria de Informática (E. U. Informática) [9] de la Universidad Politécnica de Madrid (UPM), a través de la plataforma de teleenseñanza (Moodle) [10] desarrollada en el marco del proyecto [4].

¹ De ahora en adelante *Agile Learning*

La estructura del artículo es la siguiente: el apartado 2 introduce brevemente la propuesta del proyecto *Agile Learning* de adoptar metodologías ágiles en el proceso de aprendizaje de la ingeniería del software. En el apartado 3, se explica la contribución de este artículo, mientras que el apartado 4 detalla su soporte tecnológico. Finalmente, el apartado 5 realiza una reflexión sobre las lecciones aprendidas de la implantación de esta propuesta, y el apartado 6 presenta las conclusiones y los trabajos futuros.

2 Adopción de Metodología Ágiles en el Aprendizaje

La incorporación de metodologías ágiles al ámbito académico requiere una adaptación al contexto de la enseñanza, en general, y al de la universidad, el centro y la asignatura en el que se imparte la docencia, en particular. Concretamente, el proyecto *Agile Learning* [4] se ha adoptado la metodología ágil Scrum [6, 7] y se ha adaptado al contexto de las asignaturas de Tecnologías de Desarrollo de Sistemas Web (TDSW) y Gestión de Proyectos y del Riesgo (GPRS) que se imparten en el departamento OEI [8] de la E.U. Informática [9] de la UPM. En esta sección se describen brevemente Scrum y la adaptación que se ha realizado en el proyecto para su adopción en el contexto de la enseñanza en el que se quiere implantar.

2.1 Metodologías Ágiles : Scrum

Scrum [6, 7] es un método ágil centrado en la gestión de proyectos. Por ello, resulta ser un método óptimo para el aprendizaje de técnicas de planificación y seguimiento de proyectos. La retroalimentación entre iteraciones constituye el elemento más potente de la metodología. El software es desarrollado por equipos que se auto-organizan en ciclos de corta duración (no más de 30 días) denominados *sprints*, comenzando cada uno de ellos con una planificación y finalizando con una *revisión* y *retrospectiva*. La Fig. 1 muestra, de forma simplificada, este modelo.

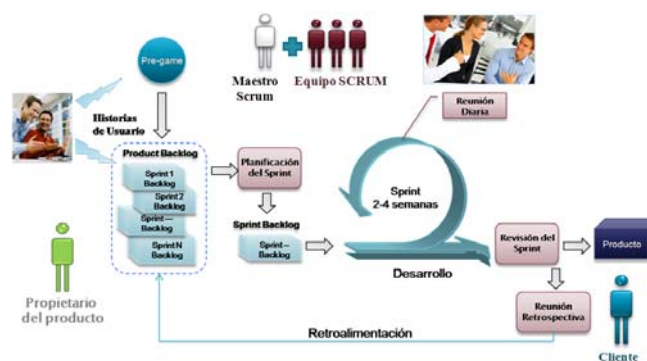


Fig. 1. Modelo de proceso Scrum

Las *historias de usuario* (requisitos) que deben ser implementadas son registradas en lo que se conoce como *backlog de producto*. El *propietario del producto* prioriza y decide cuáles son las historias de usuario más importantes que deben ser implementadas en el siguiente sprint. Los miembros del equipo coordinan sus trabajos a través de reuniones diarias de no más de 10 minutos. En ellas, uno de los miembros del equipo, conocido como el *maestro Scrum*, ayuda a solucionar los problemas que vayan surgiendo para que el trabajo sea más eficiente.

2.2 Aprendizaje Ágil

El proyecto *Agile Learning* persigue dotar al alumno, futuro ingeniero de software, del beneficio adicional de aprender y aplicar una metodología de desarrollo ágil de forma natural, adquiriendo una formación y experiencia que se le requerirá en su incorporación a la industria software. Este proyecto de innovación docente se caracterizaría por ser un proyecto:

- *Innovador*: La adopción de valores y principios ágiles junto con la aplicación de técnicas y prácticas ágiles en el ámbito de la educación resulta innovador y abre el camino hacia la agilidad.
- *Interdepartamental*: En este proyecto se involucran miembros pertenecientes dos departamentos diferentes de la E. U. de Informática de la UPM: la Unidad Docente de Empresas y el Departamento de OEI.
- *Interdisciplinar*: El proyecto abarca diferentes áreas de aprendizaje, empresa y desarrollo de software, y diferentes competencias (conocimiento, calidad técnica, trabajo en equipo, etc.)
- *Colaborativo*: Trabajo en equipos
- *Multimedia*: utilización de sistemas colaborativos web y plataformas de teleenseñanza.

Scrum, como metodología, propone un conjunto de procesos y de artefactos organizados en sprints de manera que se obtenga el máximo beneficio de la rápida respuesta del cliente al trabajo desarrollado. La implantación del modelo de proceso Scrum a la enseñanza requiere de un proceso de análisis para su adaptación al contexto específico en el que se va a implantar. Esta adaptación se debe plantear a dos niveles: por un lado la forma en que se va a trasladar el proceso Scrum al proceso de enseñanza. Y por otro, la identificación de los artefactos utilizados y del concepto de *producto final* y de *working product* durante el proceso de enseñanza.

Dentro del marco del proyecto *Agile Learning*, se han definido los conceptos de *producto final* y *working product* para, a partir de ellos, adoptar Scrum. Un *working product* en el ámbito académico es la entrega de un trabajo o un conjunto de trabajos que el profesor considera de valor como para que el alumno haya adquirido una serie de conocimientos y competencias generales evaluables. La entrega del producto final tiene como resultado final la superación de la asignatura, lo que implica que el alumno ha adquirido el conocimiento y competencias requeridas por la asignatura. Por lo tanto, el producto final lo conforma el conjunto de *working products* que el alumno entrega al profesor y son evaluados a lo largo del tiempo en el que se imparte la asignatura. El producto final se obtiene como resultado de un proceso de aprendizaje iterativo e incremental.

Dadas las particularidades del sistema educativo en relación con la inercia de funcionamiento, infraestructuras y la adopción de nuevas prácticas, metodologías, etc., no es factible la implantación completa de Scrum como base del modelo de aprendizaje en un único paso. Por ello, el proyecto *Agile Learning* define una hoja de ruta que permite esta adopción de Scrum en el ámbito educativo universitario de una forma iterativa, incremental y viable. El objetivo de la hoja de ruta es ayudar a reproducir esta experiencia en otros centros guiándoles en aquellas cuestiones que han de tener en cuenta a la hora de su implantación. Esta hoja de ruta se divide en diversas fases que se detallan a continuación (ver Fig. 2):

1. *Definición del equipo Scrum.* Los equipos Scrum están formados por un conjunto de alumnos. Estos pueden ser de diferente índole y número dependiendo de un conjunto de aspectos a tener en cuenta:
 - a. *Número de alumnos involucrados:* Se recomienda que los equipos estén formados por un número de alumnos que permita reproducir el ambiente de un grupo de trabajo. Por ello, se recomienda que se encuentre entre 4 y 6 alumnos, aunque no es preceptivo, ya que hay asignaturas que por su índole requieren un número mayor o menor.
 - b. *Trabajos a realizar:* El número y tipo de los trabajos a realizar dependerá de la naturaleza y complejidad de los trabajos a realizar. No tendrá mucho sentido crear equipos cuando todas las tareas a realizar por los alumnos sea individuales.
 - c. *Carácter del equipo:*
 - i. *Equipo transversal a diversas asignaturas:* el equipo podrá ser el mismo *para diversas asignaturas* cuando éstas permitan la definición de un único *backlog de producto* que el equipo pueda manejar y no sea relevante durante la realización de los sprints el estado en el que se encuentra el aprendizaje de una materia.
 - ii. *Equipos independientes entre asignaturas:* el equipo varía en las asignaturas en número y en los miembros que lo conforman. Esta es la situación habitual y los alumnos tendrán que manejar diferentes *backlogs de producto*.
2. *Definición de los Sprints:* Una iteración o sprint suele durar entre 2 o 4 semanas. En este caso, la temporalidad de la iteración la determinará el profesor en base a dos posibles criterios: por temas/bloque de temas o por entregas de trabajos. Se proponen estos dos indicadores, debido a que no siempre la entrega de trabajos coincide con el fin de un tema. Cada asignatura puede plantear duraciones diferentes para los *sprints* y se recomienda que, dentro de una asignatura, la duración del *sprint* se mantenga constante. Eso sí, siempre debe estar claro desde el comienzo las iteraciones que tiene la asignatura, los temas que conforman cada iteración y los entregables asociados que se evaluarán al final de la iteración.
3. *Selección de prácticas ágiles a adoptar:* La adopción ágil educativa debe ser paulatina e incremental para que siempre tanto el alumno como el profesor vea una mejora en el proceso educativo y crezca su interés por el mundo ágil de

forma natural. El cambio completo es demasiado grande para el modelo educativo universitario y podría acabar en fracaso. Por este motivo, en este artículo proponemos siempre comenzar por la organización en equipos y *sprints* y con alguna actividad que tenga que ver con la gestión de los grupos. Concretamente, y con el objetivo de intentar cubrir la necesidad de que el alumnado desarrolle sus competencias transversales [5]. En este trabajo se propone comenzar por las reuniones retrospectivas.

4. *Adopción y adaptación de las prácticas seleccionadas al contexto*: Una vez seleccionada una práctica, para poder adoptarla hay que ver cómo articular dicha actividad dependiendo del contexto del equipo Scrum. Esto radica principalmente en la cantidad de horas de trabajo que disponga el equipo Scrum para realizar trabajo conjunto in situ: existirán casos en que los alumnos trabajen además de estudiar, y hagan el trabajo a distancia de forma colaborativa, pudiendo reunirse solo de manera esporádica. Y otros en los que los equipos Scrum comparten muchas horas de trabajo y clase juntos. En base a ello se deberán de tomar unas medidas de adaptación para la correcta adopción de la actividad.

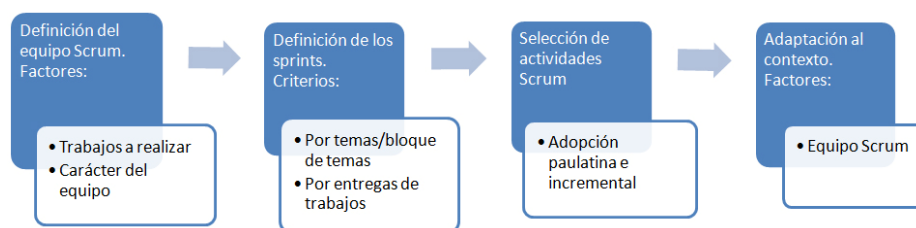


Fig. 2. Hoja de Ruta de Adopción de Agile Learning

3 Retrospectivas para el Aprendizaje de Ingeniería del SW

Desarrollo ágil significa mejorar continuamente la forma de trabajar. Las retrospectivas representan un mecanismo adecuado para comenzar a identificar las cosas que se pueden mejorar [11]. Las retrospectivas se llevan a cabo al finalizar un sprint y se centran en analizar lo que ha funcionado bien, lo que no se ha hecho correctamente y lo que se puede mejorar para la siguiente iteración. Estas duran alrededor de 2 horas y constan de las siguientes fases: descripción de la práctica, identificación de problemas, identificación de las causas, establecimiento de un plan de acción y conclusiones. En el ámbito de la enseñanza no es posible aplicar este mismo formato ya que un grupo de trabajo no suele dedicar más de 10 horas semanales a cada asignatura y dedicar 2 horas a analizar el trabajo supondría una gran sobrecarga de control. Por ello, es necesario buscar una particularización que se adapte a las características del entorno.

En el caso del proyecto *Agile Learning*, las retrospectivas consisten en una reunión en horario de tutorías con el profesor de la asignatura tras la entrega y revisión del *working product*. A esta reunión deben asistir físicamente los miembros del equipo y

se completa con una evaluación del equipo y de sus miembros que formará parte de la calificación del *working product* entregado. Esta reunión sirve para que el equipo se auto-evalúe como grupo de trabajo. De ahí que se requiera la presencia de cada uno de los miembros, y si no es posible, facilitar la conexión a distancia con algún software que permita realizar teleconferencias. En esta auto-evaluación se tratan tantos los puntos positivos y negativos, como los puntos a mejorar del funcionamiento en equipo para el siguiente *sprint* y de cada uno de los miembros individualmente. Asimismo, el profesor les evalúa indicando qué cosas deben subsanar o mejorar, y reforzar aquello que realicen de forma satisfactoria. Es importante destacar que las retrospectivas las emplea el profesor para evaluar las competencias generales del aprendizaje mientras que las competencias específicas de la misma se evalúan mediante las revisiones de los *sprints*. Esto es fundamental, ya que se ha de hacer un análisis y seguimiento de cómo cada uno de los miembros alcanza como equipo y de forma individual las competencias generales que se les requieren. De este modo, se asegura que siempre en todas las retrospectivas se analicen estos temas y se consiga realizar un seguimiento de la evolución de cada una de estas competencias generales de la asignatura.

Además de esta adaptación de la reunión retrospectiva al ámbito académico, la retrospectiva se ha extendido mediante la realización de una evaluación no pública de las competencias adquiridas después de la reunión retrospectiva. Esta evaluación privada se realiza a través de la plataforma *Moodle* adaptada en el proyecto (ver sección 4). Dicha evaluación es necesaria, ya que en el ámbito académico el buen o mal funcionamiento como equipo y como miembros de éste, va a afectar a la calificación final en la asignatura. Con el objetivo de evitar suspicacias, complicidad y/o falta de sinceridad en la calificación, la calificación se hace de forma privada. Dicha evaluación consta de tres evaluaciones: evaluación del profesor a cada miembro y al grupo, la auto-evaluación de cada miembro del equipo, y la co-evaluación de un miembro del resto de miembros del equipo. Por lo tanto, la reunión retrospectiva se redefine como se muestra en la Fig. 3.

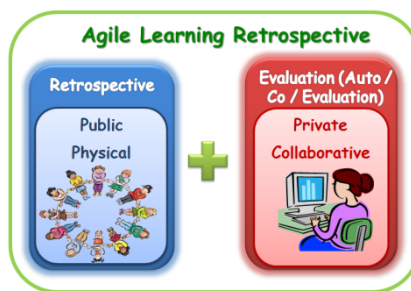


Fig. 3 Agile Learning Retrospective

La aplicación específica, en el ámbito de los estudios de la Ingeniería del Software, se realiza de forma incremental, tanto en la metodología (como ya se ha mencionado anteriormente) como en el número de asignaturas involucradas. Concretamente, se ha implantado en dos asignaturas: Tecnologías de Desarrollo de Sistemas Web (TDSW) y Gestión de Proyectos y del Riesgo (GPRS). En el caso de TDSW, la adopción se realizó sobre un subconjunto de alumnos. Se formaron tres equipos Scrum como

proyecto piloto, y en el caso de GPRS la implantación se realizó sobre el total de los alumnos de la asignatura. En el caso de TDSW fueron 3 equipos de 3 alumnos cada uno. Y en el caso de GPRS, se organizaron 21 equipos Scrum formados por 4, 5 o 6 alumnos, siendo mayoritariamente grupos de 5 alumnos. La naturaleza de los equipos Scrum era de una gran variedad, en muchos casos todos los componentes trabajaban y estudiaban simultáneamente, y por tanto el trabajo normalmente se hacía de forma colaborativa y distribuida, con los retos que esto implica para el trabajo en equipo [12].

4 Moodle Agile Learning

Todo este soporte ágil para el aprendizaje y evaluación de asignaturas de ingeniería del software y por ende, aprendizaje de metodologías ágiles en un contexto académico, requiere de una gestión organizada, debido al volumen de equipos que se han de gestionar de forma simultánea y sincronizada. Para ello, se ha especializado una plataforma de enseñanza *Moodle* [10] para la construcción una plataforma específica con las características requeridas para implantar este nuevo concepto de aprendizaje ágil. Esta plataforma se conoce como *Agile Moodle* [4]. En esta sección se va a ilustrar cómo dicha herramienta da soporte a la definición de iteraciones, equipos y finalmente, retrospectivas.

La herramienta está orientada al trabajo colaborativo basado en grupos de trabajo. A cada grupo se le ofrece un conjunto de herramientas colaborativas para facilitar el desarrollo de su trabajo. Estas herramientas solo tienen visibilidad a nivel de grupo lo que les permite aislar los contenidos y asegurar la privacidad de sus contenidos. En cuanto a la organización de la asignatura, los contenidos se estructuran en iteraciones. En cada iteración, se definen tareas y trabajos a desarrollar, y se pone a disposición de los equipos el material y documentación necesario para adquirir los conocimientos específicos de la asignatura. Por ejemplo, en la asignatura de GPRS se definieron un total de 6 iteraciones. En las que se proporciona el material que se imparte, el conjunto de entregables que conforman el *working product* de la iteración, y finalmente las reuniones, entre ellas retrospectiva (ver Fig. 4).

El calendario de la iteración es el que establece la fecha para la reunión retrospectiva física. Una vez celebrada la reunión física, se habilita la actividad llamada retrospectiva dentro de la plataforma para que los alumnos y el profesor puedan realizar la evaluación, auto-evaluación y co-evaluación del equipo durante esa iteración en base a las conclusiones de la reunión retrospectiva física. En esta evaluación privada, vía la plataforma *Agile Moodle*, se evalúan las competencias generales objeto de la asignatura. Por ejemplo, en la asignatura de GPRS se evaluaron las competencias: Capacidad de Análisis, Capacidad de Organización, Trabajo en Equipo, Razonamiento Crítico, y Liderazgo. Dicha evaluación se realiza mediante una tabla en la que se permiten valorar del 1 al 10 cada una de las competencias generales a cada uno de los miembros del equipo, obteniendo una nota de su auto-evaluación, co-evaluación, y evaluación del profesor de las competencias a nivel individual y en equipo (ver Fig. 5). Finalmente, se obtienen informes que permiten

analizar y realizar un seguimiento de la evolución de adquisición de competencias del equipo y de sus miembros (ver Fig. 6).

Iteraciones

- Iteración 0
- Iteración 1. Semanas 1, 2 y 3
- Iteración 2: Semanas 4, 5 y 6
- Iteración 3: Semanas 7 y 8
- Iteración 4: Semanas 9, 10 y 11
- Iteración 5: Semanas 12, 13 y 14
- Iteración 6: Semanas 15 y 16
- Práctica 1
- Práctica 2
- Actividad complementaria
- DCAL

1 Iteración 0
Establecimiento de las infraestructuras de la asignatura
Asignación de casos de estudio a los grupos de clase

2 Iteración 1. Semanas 1, 2 y 3

Semana 1
Tema 1: Concepto de proyecto informáticos

Semana 2
Tema 2: Roles que intervienen en el desarrollo de proyectos informáticos
Tema 3: Etapas en el desarrollo de proyectos informáticos

Semana 3
Tema 4: Herramientas para la gestión de proyectos: Formalización

Working product

- Definición del ámbito del caso de estudio
- Definición de los roles del caso de estudio
- Definición de las etapas del proyecto para el caso de estudio
- Formalización del caso de estudio

Material

- Trasparencias Tema 1. Enfoque de proyecto
- Trasparencias Tema 1. Ciclos de Vida
- Trasparencias Tema 1. Requisitos
- Trasparencias Tema 2. Gestión de equipos
- Trasparencias Tema 3. Etapas de un proyecto
- Trasparencias Tema 4. Formalización de proyectos

Retrospectiva

- Revisión Iteración 1
- Asignación Roles Iteración
- Retrospectiva Iteración 1

Fig. 4. Definición de iteraciones: material, *working product* y reuniones

Ver los informes

Grupos separados: GSM31-G1

Evaluación del trabajo del equipo para la elaboración de la segunda entrega de la práctica de ágiles

	Alumno1	Alumno2	Alumno3	Alumno4	Grupo
Capacidad de Análisis	90	80	90	90	90
Capacidad de Organización	90	80	90	90	90
Trabajo en Equipo	90	90	90	90	90
Razonamiento Crítico	90	80	80	90	90
Liderazgo	90	90	90	90	90

Guardar mi elección

Fig. 5. Retrospectiva de Evaluación de Competencias Generales

Grupos separados GSM31-G1

Capacidad de Análisis

Evaluación de grupo	Alumno1	Alumno2	Alumno3	Alumno4	Grupo
Alumno1	90	90	90	90	90
Alumno2	100	90	100	100	100
Alumno3	90	90	80	100	90
Alumno4	90	80	80	90	90
Evaluación del Profesor					
Profesor	90	80	90	90	90

Fig. 6. Informe de la Capacidad de Análisis de la Retrospectiva de la Iteración 5

5 Lecciones aprendidas

Esta sección presenta de forma resumida las primeras lecciones que se han consolidado a la finalización de los cursos de las asignaturas TDSW y GPRS. Las lecciones aprendidas se presentan en dos grupos: sobre las metodologías ágiles y sobre las materias específicas de Ingeniería del software.

5.1 Aprendizaje de Metodologías Ágiles

Debido a que los alumnos no estaban familiarizados con Scrum ni con las prácticas ágiles, fue necesaria una sesión inicial de presentación de la metodología. Los equipos de GPRS necesitaron aproximadamente tres iteraciones para asimilar la nueva forma de trabajar y alcanzar de forma regular los objetivos de las iteraciones.

Durante las primeras iteraciones, los grupos tenían un comportamiento más cercano a estructuras jerárquicas de funcionamiento en cuanto a la asignación de tareas. Al final del curso, la mayoría de los grupos, trabajaban de forma proactiva, lo que aumentaba la motivación en la realización de los trabajos. También durante las primeras iteraciones, las retrospectivas reflejaban problemas de asimilación de la metodología. Los problemas de funcionamiento detectados estaban directamente relacionados con los propios procesos de Scrum, según avanzaba el curso, los problemas se fueron centrandos en la resolución de conflictos internos de funcionamiento y no tanto de la adopción. Un aspecto relevante de las conclusiones emitidas por los participantes en el proyecto es que tenían la sensación de tener las materias más controladas, fundamentalmente por el seguimiento y la adopción de Scrum como forma de trabajo. También se detectaron algunos problemas, sirvan como ejemplo: falta de participación de algunos alumnos, absentismo o problemas para compaginar la dedicación exigida con otras materias impartidas de forma convencional. Algunos de estos problemas no pudieron ser resueltos, incluso a pesar del papel activo que los profesores tuvieron para buscar soluciones. Si bien no puede hacerse responsable a la metodología de la aparición de estos problemas, tampoco permitió encontrar una solución para los mismos.

5.2 Aprendizaje de Ingeniería del Software

Con respecto a los contenidos específicos de las materias incorporadas al proyecto, pueden extraerse las siguientes conclusiones. La utilización de metodologías ágiles no ha representado una mejora en el aprendizaje de contenidos con respecto a los resultados de años anteriores, al menos en la asignatura de TDSW. Es decir, los conocimientos adquiridos son los mismos. Sin embargo, el hecho de realizar con ellos reuniones periódicas de revisión de *working product* les hacía llevar al día los trabajos, algo que también ocurre cuando los alumnos están inmersos en asignaturas guiadas por evaluación continua. Esto ha implicado que el índice de éxito en las asignaturas haya sido alto, si bien no puede extrapolarse a que en todos los casos de aplicación puedan obtenerse resultados semejantes. Con respecto a los contenidos, al ser evaluados periódicamente, aquellos alumnos con menor disponibilidad temporal fueron los que no consiguieron resultados satisfactorios. Esto puede apuntar que la aplicación de este tipo de metodologías en la enseñanza no sea la más adecuada cuando los estudiantes tienen que compaginar el estudio con otro tipo de actividades (laborales, familiares, etc.).

Con respecto a la adquisición de competencias generales, los miembros del grupo de trabajo han podido comprobar que ya solo el hecho de indicar a los alumnos que van a ser evaluados en dichas competencias, provoca en ellos una reacción que les lleva a cuidar mucho más todos los aspectos que tienen que ver con dichas competencias y por tanto, una mejora en la disposición a practicar con corrección las habilidades que se les proporcionan. En el marco del proyecto se han tenido en cuenta aspectos como la "dirección" o liderazgo y organización. Por otra parte, el hecho de trabajar en equipo hace que las calificaciones individuales sean más complicadas de obtener por parte de los docentes, ya que al no existir una prueba final individual, es complejo diferenciar el nivel de conocimientos alcanzado por cada miembro del equipo. Por último hay que destacar que los aspectos relacionados con la obtención de calificaciones cruzadas entre alumnos (co-evaluaciones) siempre aportan una serie de compromisos entre ellos que pueden provocar que dichas evaluaciones no sean nunca negativas a no ser que en los equipos reine el caos más absoluto como forma organizativa

6 Conclusiones y Trabajos Futuros

Hoy en día, el desarrollo ágil de software ha adquirido una gran relevancia en el área de la industria software y hay un gran interés en la adopción de las metodologías ágiles. En este artículo se propone como primer paso hacia la agilidad, el que los ingenieros de software que se incorporen al mundo laboral tengan las habilidades y conocimientos para poner en marcha metodologías ágiles. Para ello, se hace necesario que en las titulaciones de Ingeniería del Software se impartan estas metodologías. Sin embargo, en este artículo se da un paso más adelante, y se apuesta por una puesta en marcha de las metodologías durante la formación del ingeniero. De este modo, su preparación será mayor y su adaptación al mundo laboral ágil será instantánea.

En este artículo se presentan los primeros pasos para la adopción de metodologías ágiles en el ámbito académico como resultado de la implantación del proyecto *Agile*

Learning. En línea con el proyecto, este artículo presenta la experiencia de impulsar los principios y valores en los que se basan las metodologías ágiles para obtener dos ventajas fundamentales: su aprendizaje y el fomento de la adquisición de competencias generales como la capacidad de organización, el trabajo en equipo, la comunicación, o el liderazgo, entre otras. Concretamente, este artículo presenta una guía de implantación del trabajo en equipos y reuniones retrospectivas para que pueda ser reproducida dicha adopción por otras instituciones. Además, presenta el soporte tecnológico necesario para esta adopción, una plataforma de tele-enseñanza especializada para las necesidades planteadas.

Este trabajo abre un camino muy largo que recorrer con una gran de trabajos futuros, entre los que cabe destacar la adaptación de cada una de las técnicas y prácticas ágiles al ámbito académico y la mejora y construcción de mecanismos tecnológicos que lo faciliten y lo hagan viable. Este es el primer paso hacia la agilidad.

Agradecimientos

Este trabajo está cofinanciado por el proyecto de innovación educativa Aprendizaje Ágil- *Agile Learning* perteneciente al proyecto de innovación educativa de la E.U. Informática de la UPM y el proyecto *INNoSEP: INcorporating inNOvation in Software Engineering Processes* TIN2009-13849. Además, los autores quieren hacer un agradecimiento a los miembros del proyecto *Agile Learning*, y en especial a los profesores Jorge Tejedor y F. Javier Gil por el esfuerzo realizado en la implantación de estas ideas en la asignatura TDSW junto a los autores de este artículo. Finalmente, agradecer a Pablo Ortíz el trabajo realizado en el desarrollo de la plataforma Agile Moodle que ha permitido hacer realidad dicho proyecto.

Bibliografía

1. Boehm, B.: A View of 20th and 21st Century Software Engineering. In: 28th international conference on Software engineering, pp. 12--29. Shanghai, China, 2006.
2. Agile Manifesto, 2011, <http://agilemanifesto.org/>
3. Ambler S.: Agile Adoption Survey February 2008 www.agilemodeling.com/surveys
4. Plataforma Moodle Agile Learning, <http://agilelearning.eui.upm.es/>
5. Díaz, J., Pérez, J., Yagüe J., Alonso, S., Gil J.: Análisis Empírico del Papel de las Competencias Generales en el Marco de los Estudios Superiores, XVII Jornadas de Enseñanza de la Informática (JENUI 2011), Julio 2011.
6. Schwaber K.: Agile Project Management with Scrum, Microsoft Press, 2004.
7. Schwaber K. and Beedle M., Agile Software Development with Scrum, Prentice Hall, 2001.
8. Departamento de Organización y Estructura de la Información (OEI), <https://srvoei.eui.upm.es/web/>
9. Escuela Universitaria de Informática, <http://www.eui.upm.es>
10. Moodle, <http://moodle.org/>
11. Crispin, L., Gregory, J.: Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley Professional, 2009.
12. Eckstein J.: Agile Software Development with Distributed Teams. Dorset House Publ. Co., Inc., New York, NY, USA 2010.