

Experiencia y Lecciones Aprendidas durante el Desarrollo de un Proyecto Software Común a Diversas Asignaturas*

Pablo Sánchez¹, Carlos Blanco¹, Alejandro Pérez¹, Julio Medina¹, Patricia López¹, Alfonso de la Vega¹, Diego García¹ y Miguel Sierra^{1,2}

¹Dpto. de Ing. Informática y Electrónica, Universidad de Cantabria, Santander; ²CIC Consulting Informático
{sanchezbp, blancobc, perezruiza, medinajl, lopezpa, delavegaa, garciasad}@unican.es, m.sierra@cic.es

Resumen

Las asignaturas de un plan de estudios, a pesar de mantener claras dependencias y relaciones entre ellas, suelen gestionarse e impartirse como unidades aisladas e independientes. Esto genera una serie de problemas, como la falta de una visión conjunta del conocimiento, que pueden afectar negativamente al aprendizaje. Con objeto de reducir estos posibles problemas, e inspirados por otros autores, tratamos de romper las barreras artificiales existentes entre asignaturas mediante su integración a través del desarrollo de un proyecto común. Este trabajo describe dicha experiencia, las decisiones adoptadas y los resultados obtenidos con objeto de que pueda servir de guía e inspiración a otros compañeros.

Abstract

Subjects in a curriculum are usually managed and taught as independent isolated units, even though they are often clearly inter-dependent and inter-related. This creates some problems, such as the lack of a general perspective of the matter, which might hinder learning. Inspired by other authors, and in order to improve the educational experience, we try to break the artificial borders between subjects by integrating them through the development of a common project. In the aim that this may serve as guidance and inspiration to others, this work describes that experience, the decisions we made, and the results obtained.

Palabras clave

Integración de Asignaturas. Enseñanza por Materias. Aprendizaje Basado en Proyectos. Ingeniería del Software. Calidad Software. Pruebas Software. Scrum.

*Este trabajo ha sido parcialmente financiado por la III Convocatoria de Proyectos de Innovación Docente de la Universidad de Cantabria y el proyecto TIN2014-56158-C4-2-P.

1. Introducción

Las asignaturas de un mismo plan de estudios, salvo casos muy excepcionales, no constituyen unidades independientes de conocimiento, sino que mantienen fuertes vínculos entre sí. Sin embargo, tal como ya han constatado diversos autores [1, 2, 3, 8, 9], estas asignaturas interdependientes suelen gestionarse como entidades aisladas, tanto desde un punto de vista administrativo como del aprendizaje.

Este aislamiento artificial afecta negativamente al aprendizaje de los estudiantes. Por ejemplo, tal como exponen Barrado et al. [1], cuando se emplea aprendizaje basado en proyectos [5], la excesiva fragmentación del conocimiento dificulta el desarrollo de proyectos ambiciosos dentro del ámbito de una única asignatura.

Para resolver este y otros problemas, decidimos tratar de romper las barreras actualmente existentes entre las asignaturas de la intensificación en Ingeniería del Software del Grado en Ingeniería Informática de nuestra universidad. Para ello, inspirados por las experiencias previas y los resultados positivos de otros autores [1, 2, 3, 8, 9], nos planteamos la posibilidad de desarrollar un proyecto común entre varias de estas asignaturas.

Durante el curso 2015/2016 exploramos la viabilidad de esta idea mediante la integración de las asignaturas de *Métodos de Desarrollo y Calidad y Auditoría*. Durante una parte del cuatrimestre, el objetivo de ambas asignaturas fue desarrollar un único proyecto, común a las dos asignaturas. Todas las horas de aula de estas asignaturas se dedicaron al desarrollo de dicho proyecto. En cada hora, los alumnos podían trabajar en actividades de una u otra asignatura a su conveniencia, con independencia de a qué asignatura concreta correspondía realmente cada hora. Por tanto, durante esa parte del cuatrimestre, las asignaturas fueron indistinguibles.

Animados por los buenos resultados obtenidos, durante el curso 2016/2017 sumamos la asignatura de

Procesos de Ingeniería del Software a esta iniciativa, repitiéndose los buenos resultados. Este proceso de integración no está aún cerrado, pues en futuros cursos nos gustaría sumar nuevas asignaturas a este proyecto.

Este trabajo describe nuestra experiencia hasta el momento, los problemas que hemos encontrado, las decisiones que adoptamos y los resultados que obtuvimos. Nuestro objetivo es animar a otros compañeros a emprender acciones similares y que nuestra experiencia pueda servirles de apoyo durante su puesta en marcha.

Tras esta introducción, este trabajo se estructura del siguiente modo: el apartado 2 describe los problemas concretos que pretendíamos resolver mediante la integración de las asignaturas. El apartado 3 proporciona una visión general de la solución que proponemos; mientras que el apartado 4 detalla cómo se implementó dicha solución. El apartado 5 analiza de manera crítica los resultados obtenidos, sirviendo el apartado 6 de resumen y cierre a este trabajo.

2. Problemas a resolver

Este trabajo se enmarca dentro de la intensificación en Ingeniería del Software del Grado en Ingeniería Informática de la Universidad de Cantabria. Dicha intensificación está compuesta por 8 asignaturas que se cursan al final del Grado y que son obligatorias para los alumnos que eligen dicha intensificación. Cada asignatura tiene una carga de 6 créditos ECTS (*European Credit Transfer System*).

Los problemas concretos que pretendíamos resolver, y que estaban relacionados con la fragmentación del conocimiento dentro de estas asignaturas, se describen a continuación.

P1. Presencia de actividades redundantes. En primer lugar detectamos que el aislamiento entre asignaturas provocaba que ciertas actividades tuvieran que repetirse en diversas asignaturas. Por ejemplo, tanto en la asignatura de *Métodos de Desarrollo* como en la de *Calidad y Auditoría* se llevaban a cabo actividades para el control de la calidad de un producto software. Por otro lado, tres asignaturas distintas, *Métodos de Desarrollo*, *Calidad y Auditoría* y *Procesos de Ingeniería del Sw*, requerían de la definición e implementación de casos de prueba.

Estas repeticiones hacían que cada asignatura tuviese que dedicar un cierto tiempo a realizar y evaluar actividades relacionadas con competencias de otras asignaturas. Por tanto, el tiempo que cada asignatura puede dedicar a desarrollar sus propias competencias específicas es menor, reduciéndose su nivel de aprendizaje.

Además, estas redundancias incrementaban el riesgo de la aparición de inconsistencias, ya que cada asignatura,

al perseguir objetivos diferentes, podía utilizar criterios de evaluación diferentes. Estas diferencias podrían confundir al alumnado, que en algunos casos podría no tener claro qué debía hacer exactamente. Esta inseguridad puede generar un cierto estado de ansiedad y estrés que no es nada favorable ni para el aprendizaje ni para la relación alumno-profesor.

P2. Proyectos poco atractivos. Tal como señalan Barrado et al. [1], el tamaño actual de las asignaturas, en torno a los 6 créditos ECTS (*European Credit Transfer System*), dificulta la realización de proyectos de cierta envergadura dentro de una única asignatura. Si además, tal como hemos comentado antes, cada asignatura tiene que dedicar parte de su tiempo a desarrollar actividades relacionadas con otras asignaturas, el problema se agrava.

El resultado es que es muy complejo desarrollar un proyecto completo, de principio a fin, dentro de una única asignatura. Por tanto, estos proyectos se desarrollan con frecuencia sólo parcialmente, por lo que rara vez se obtiene como resultado un producto tangible. Esta carencia de resultados visibles es frustrante para el alumno.

Además, estos proyectos pequeños, incompletos, de escasa complejidad, y en ocasiones artificialmente modificados para adecuarlos al contexto de una asignatura, suelen ser percibidos por los alumnos como meros ejercicios académicos, alejados de la práctica profesional. Dicha percepción puede afectar negativamente a su motivación [1].

P3. Carencia de una visión integradora del conocimiento. La fragmentación del conocimiento en asignaturas, tal como ha sido estudiado en diversos trabajos [1, 9, 3, 2], puede provocar que las conexiones entre asignaturas no se trabajen de manera adecuada y, por tanto, se pierdan. Consecuentemente, los alumnos podrían tener problemas a la hora de aplicar lo aprendido en varias asignaturas de manera conjunta dentro de un mismo proyecto.

Este hecho se debe a que, como hemos comentado anteriormente, cada asignatura tiene sus propios objetivos de aprendizaje y no mucho tiempo para desarrollarlos. Por tanto, cada asignatura se centra en los aspectos que son más relevantes para alcanzar sus objetivos, ignorando el resto de elementos que los rodean, o adquiriendo visiones bastante simplistas de los mismos. Por ejemplo, una asignatura podría centrarse en producir informes de calidad pero no detallar cómo afectan esos informes al desarrollo del producto.

Este hecho concreto nos fue indicado por diversas empresas del sector durante una reunión que mantuvimos con ellas para analizar el nivel de nuestros egresados. Estas empresas comentaron que mientras la for-

mación técnica de nuestros alumnos sí era bastante buena, detectaban bastantes carencias en su formación metodológica, ya que los recién licenciados exhibían dificultades para adaptarse a sus procedimientos de trabajo.

3. Solución propuesta

Para resolver los problemas anteriores, nuestra primera iniciativa fue la de tratar de eliminar las redundancias entre actividades, procurando que cada actividad se realizara sólo una vez y que esta realización fuese compartida por varias asignaturas. Esto resolvería nuestro primer problema (P1) y ayudaría a disminuir el riesgo de inconsistencias. Además, al reducir actividades, dispondríamos de más tiempo para abordar los proyectos, lo que contribuiría a resolver nuestro segundo problema (P2).

Siguiendo los consejos de Villagrà et al. [9], para llevar a cabo esta integración era recomendable disponer de un elemento que sirviese de eje de unión para los diversos conocimientos. Este elemento ayudaría además a resolver el problema de clarificar cómo se integran conocimientos de asignaturas diversas, que era nuestro tercer problema (P3).

En nuestro caso, dicho elemento vertebral fue el desarrollo de un proyecto conforme a la metodología *Scrum* [7]. Esta actividad abarcaba el desarrollo de un proyecto software de manera casi completa e implicaba la utilización de conocimientos procedentes de diversas asignaturas. Dicha actividad se realizaba dentro de la asignatura de *Métodos de Desarrollo*.

Una vez escogido nuestro elemento vertebral, analizamos qué asignaturas concretas tenían actividades solapadas con él. Para facilitar el proceso de integración, decidimos seleccionar asignaturas del mismo cuatrimestre, ya que la integración de asignaturas de diferente cuatrimestre implica siempre una mayor complejidad. Las asignaturas que compartían cuatrimestre con *Métodos de Desarrollo* eran *Calidad y Auditoría*, *Procesos de Ingeniería del Sw* y *Diseño Sw*. Los objetivos de aprendizaje de cada una de estas asignaturas se muestran de manera resumida en el Cuadro 1.

Durante el curso 2015/2016, hicimos una prueba piloto de nuestra idea e integramos las asignaturas de *Calidad y Auditoría* y *Métodos de Desarrollo*. Al año siguiente, durante el curso 2016/2017, se amplió la integración a la asignatura de *Procesos de Ingeniería del Sw*. La selección de estas asignaturas se basó tanto en la predisposición inicial de sus profesores a participar en el proyecto como en la facilidad para integrar dichas asignaturas en el proyecto.

El desarrollo del proyecto en *Scrum* cubría el objetivo MD.2, y requería de la utilización de técnicas para la gestión de la configuración (MD.1), realización de

| Objetivos de Aprendizaje | |
|--------------------------------|---|
| El alumno deberá ser capaz de: | |
| MD | 1. Gestionar la configuración de un producto sw. 2. Desarrollar sw conforme a una metodología ágil. 3. Modelar procesos de desarrollo sw. |
| CA | 1. Controlar la calidad de un producto sw. 2. Auditar productos y procesos sw. |
| PISw | 1. Desarrollar pruebas para un producto sw. 2. Trabajar con plataformas empresariales (JEE). |

MD: Métodos de Desarrollo; CA: Calidad y Auditoría; PISw: Procesos de Ingeniería del Sw.

Cuadro 1: Objetivos de Aprendizaje.

pruebas (PISw.1) y control de la calidad (CA.1). Además, el control de calidad de un producto sw (CA.1) requería de la existencia previa de unas pruebas para dicho producto (PISw.1). Por tanto, la idea era desarrollar estos objetivos mediante la realización de actividades comunes a las tres asignaturas.

Sin embargo, no todos los objetivos de las tres asignaturas tenían cabida dentro de un proceso de desarrollo. Por ejemplo, el modelado de un proceso software (MD.3) no es algo que se haga durante el desarrollo de un producto software, sino durante la definición de los métodos de trabajo que seguirá una determinada empresa. De modo parecido, las auditorías (CA.2) suelen realizarse sobre productos terminados, y no durante su desarrollo.

En un principio planteamos la idea que el proyecto a desarrollar fuese una aplicación empresarial desarrollada utilizando la tecnología JEE (*Java Enterprise Edition*), con lo que cubriríamos también el objetivo PISw.2. Sin embargo, nuestra experiencia previa nos aconsejaba no utilizar proyectos con una alta complejidad tecnológica, ya que, en esos casos, los alumnos tendían a centrarse en los problemas técnicos descuidando las cuestiones metodológicas. Una especial atención a las cuestiones metodológicas era clave para resolver nuestro tercer problema (P3). Por tanto, decidimos dejar fuera del proyecto también este objetivo.

Merece la pena resaltar que, a diferencia de otros autores [1, 2, 4], nosotros no planteábamos una integración total de las asignaturas, sino parcial. Las asignaturas se integrarían sólo duramente una parte del cuatrimestre y para el desarrollo de unos objetivos de aprendizaje muy concretos. Durante el resto del cuatrimestre se gestionarían como asignaturas separadas. Se trataba por tanto de una especie de matrimonio de conveniencia con unos objetivos muy concretos, que además no debía afectar a la consecución de los otros objetivos de cada asignatura.

Por último, necesitábamos encontrar un tema para el proyecto que fuese cautivador, lo que nos permitiría resolver el problema P2. Tras barajar varias opciones, optamos por el desarrollo de una pequeña aplicación

para Android que se conectase a un servicio externo de datos y permitiese navegar de manera cómoda por los datos recuperados. La utilización de un servicio externo de datos real ayudaba a dar veracidad al proyecto. Además, el desarrollo de aplicaciones para móviles era algo que motivaba bastante a los alumnos y no se cubría en ninguna asignatura de la carrera. Por tanto, esperábamos que ambos elementos nos ayudaran a resolver el problema P2.

Respecto a la complejidad tecnológica antes mencionada, hay que destacar que el desarrollo de aplicaciones para Android, salvo por cuestiones de interfaz gráfica y siempre y cuando no se usen funciones avanzadas de los teléfonos móviles, no difiere mucho del desarrollo de aplicaciones Java estándar. Por tanto, no revestía una gran complejidad tecnológica. Además, contábamos con expertos en Android entre los profesores involucrados en el proyecto, por lo que era factible impartir unos pequeños seminarios sobre el tema a comienzo del curso.

Por tanto, el problema concreto que debíamos resolver era el de cómo integrar tres asignaturas concretas, *Métodos de Desarrollo*, *Procesos de Ingeniería del Software* y *Calidad y Auditoría*, de manera que, durante parte de un cuatrimestre y con unos objetivos de aprendizaje muy concretos, se desarrollase entre las tres una aplicación Android utilizando la metodología Scrum. El siguiente apartado describe cómo llevamos a cabo dicho reto.

4. Desarrollo de un proyecto software integrado usando Scrum

El objetivo principal de Scrum [7] es verificar de manera temprana que el producto que se está desarrollando realmente se ajusta a lo deseado, de manera que, en caso de no ser así, se puedan adoptar acciones correctivas lo más pronto posible, evitando rehacer grandes cantidades de trabajo. Para alcanzar este objetivo, los productos software se desarrollan a través de iteraciones de corta duración, denominadas *sprints*, al final de las cuales se entrega una nueva versión operativa del producto. Estas versiones operativas se utilizan para comprobar que lo desarrollado coincide con lo deseado.

Dentro de este esquema general, Scrum define una serie de roles y un conjunto pequeño pero concreto de actividades a desarrollar en cada sprint (ver Figura 1). Los siguientes subapartados describen cómo dichos roles y actividades se representaron dentro de nuestro proyecto.

4.1. Roles en Scrum

Dentro de un proyecto Scrum se distinguen los siguientes roles:

Scrum Team. Es el equipo que desarrolla el producto. Dentro de un equipo Scrum, no se distinguen roles específicos, como programador o ingeniero de pruebas, sino que cualquier miembro debería ser capaz de realizar cualquier función.

Product Owner. Es el representante de los *stakeholders* y el responsable de la dirección estratégica del producto. Verifica que el producto se desarrolla conforme a lo solicitado y decide el orden en el cual se irán desarrollando y publicando sus funcionalidades.

Scrum Master. Es un experto en Scrum que vigila que se sigan los principios de la metodología. Su función es la de resolver dudas sobre cuestiones metodológicas y tratar de mejorar el funcionamiento de los equipos.

Para la creación de los *Scrum Teams*, dividimos a los alumnos en grupos. Scrum recomienda un tamaño para los equipos de entre 4 y 10 personas. Nuestra experiencia aconseja crear grupos de alrededor de 6 alumnos. Con menos alumnos es complicado desarrollar funcionalidades importantes dentro de un único sprint; mientras que con un tamaño superior se puede complicar demasiado la gestión del grupo.

A cada grupo se le asignó un profesor distinto que ejercería el rol de *Product Owner*. Por último, un profesor asociado, con certificación oficial de *Scrum Master* y experiencia en la dirección de proyectos, actuaba de *Scrum Master* para todos los grupos. Sus experiencias particulares ayudaron a los alumnos a entender bastante mejor el por qué de ciertas acciones.

4.2. Configuración de los sprints

4.2.1. Cuestiones preliminares

Basándonos en nuestra experiencia previa, decidimos que no era bueno mezclar clases de aula teóricas con el desarrollo de los sprints. En primer lugar, para poder desarrollar funcionalidades no triviales dentro del ámbito de un único sprint, necesitábamos de bastantes horas de trabajo, así que cuantas más horas se le dedicasen al proyecto, mejor. En segundo lugar, preveíamos que los alumnos centrarían todos sus esfuerzos en la entrega de una nueva versión operativa del software al final del sprint, lo que dificultaría su concentración en otro tipo de actividades.

De acuerdo con estas ideas, todas las horas disponibles de las asignaturas de *Calidad y Auditoría* y *Métodos de Desarrollo* se dedicaron al desarrollo del proyecto, no impartándose ninguna clase teórica de estas

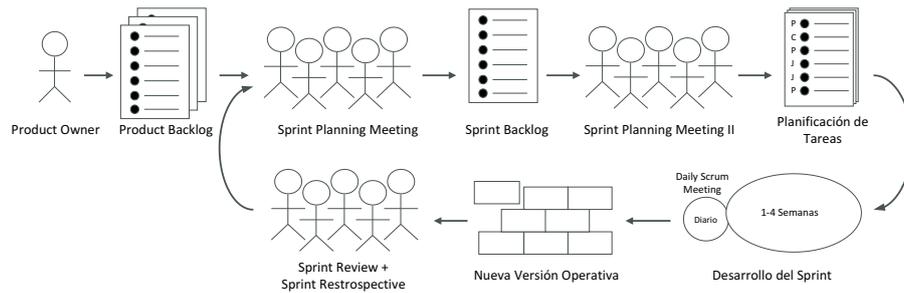


Figura 1: Metodología Ágil Scrum.

| L | M | X | J |
|----------|-----------|-----------|-----------|
| Pr (CA) | Pr (CA) | Pr (MD) | Pr (MD) |
| Pr (CA) | Pr (PISw) | Pr (CA) | Diseño Sw |
| Pr (MD) | Diseño Sw | PISw | PISw |
| Pr (MD) | Diseño Sw | PISw | Optativa |
| Optativa | Optativa | Diseño Sw | Optativa |

Pr: Proyecto Integrado; MD: Métodos de Desarrollo; CA: Calidad y Auditoría; PISw: Procesos de Ingeniería del Sw

Cuadro 2: Horario durante la realización de los sprints.

asignaturas durante su duración. Sin embargo, los profesores de la asignatura *Procesos de Ingeniería del Sw* optaron por una estrategia más conservadora, dedicando sólo una hora a la realización del proyecto. Esta fue una mala decisión, tal como analizaremos en el apartado de evaluación.

El horario de los alumnos durante el desarrollo del proyecto quedó configurado tal como se muestra en el Cuadro 2. Al lado de cada hora dedicada al proyecto se muestra la asignatura que la aportaba. Para facilitar el trabajo continuado en el proyecto, solicitamos a la dirección del centro que las horas de las asignaturas participantes fuesen contiguas.

El siguiente punto a decidir era el de la duración de los sprints. Scrum recomienda una duración para los sprints de entre 1 y 4 semanas. En nuestro caso, una semana era poco tiempo para poder conseguir una nueva versión operativa del producto. Por otro lado, sprints más largos de dos semanas, al no tener una fecha de entrega cercana, podían provocar una excesiva relajación de los alumnos al principio del sprint. Por tanto, optamos por utilizar sprints de dos semanas.

4.2.2. Diseño de las actividades

Las actividades que conformaban la metodología Scrum, ilustradas en la Figura 1, se desarrollaban de la siguiente forma. Al inicio de cada sprint, primer lunes a primera hora, cada *Scrum Team* y su correspondiente *Product Owner*, durante el *Sprint Planning Meeting I*, analizaban el *Product Backlog* para decidir cuáles serían las siguientes funcionalidades o casos de uso a

desarrollar. El *Product Backlog* es una especie de repositorio dónde se almacenan todas las funcionalidades que deberá contener el sistema. Las siguientes funcionalidades a desarrollar determinaban lo que en *Scrum* se conoce como *Sprint Backlog*.

Una vez constituido el *Sprint Backlog*, cada equipo, durante la actividad *Sprint Planning Meeting II*, descomponía el desarrollo de cada funcionalidad en tareas; se estimaba el esfuerzo de cada tarea en horas y, por último, se repartían las tareas entre los miembros del equipo. En nuestro caso, estas tareas debían incluir tanto el desarrollo de pruebas como la realización de controles de calidad.

Los elementos del *Sprint Backlog* tenían que estar completamente desarrollados para el Jueves de la siguiente semana, ya que en Scrum las fechas de finalización de un sprint son improrrogables. Para tratar de asegurar la fechas de entrega, Scrum utiliza la siguiente técnica. A cada funcionalidad a desarrollar se le asocian unos puntos que representan el esfuerzo que se estima que requerirá. Cada equipo tiene asociado un valor, denominado *velocidad de equipo*, que establece el número máximo de puntos que puede aceptar por sprint, sin poner en peligro la fecha de entrega. Esta característica resulta muy útil para controlar la cantidad de trabajo que realizan los alumnos por sprint.

Para computar la carga de trabajo semanal que debían realizar los alumnos, y poder determinar así la velocidad de los equipos, se sumaban las horas de trabajo tanto presencial como autónomo que cada alumno debía dedicar a cada asignatura integrada en el proyecto. A continuación, se aplicaba un factor de corrección para reservar parte de esas horas a tutorías, estudio y resolución de incidencias. Por cada alumno, sólo se computaban las horas de las asignaturas en las que estaba matriculado. Por ejemplo, si un alumno no estaba matriculado en *Calidad y Auditoría*, la carga de trabajo, tanto del grupo como del alumno, se veía reducida en las horas correspondientes, que eran 8 horas por semana en este caso. De este modo permitíamos la presencia en el proyecto tanto de repetidores como de alumnos que, por los motivos que fuese, no podían matricularse de la totalidad de sus asignaturas.

La asignación de tareas de cada grupo quedaba registrada en una herramienta para la gestión de proyectos Scrum (*Scrumdesk*), donde era visible tanto para los miembros del grupo como para el profesor. Por tanto, se fomentaba tanto la *interdependencia positiva* como la *exigibilidad individual*, ya que todos los alumnos se necesitaban para poder finalizar el sprint sin agobios, pero cada uno tenía sus responsabilidades individuales.

Tras la división del trabajo en tareas, cada equipo podía proceder como considerase mejor, con independencia de la asignatura a la que correspondiese cada hora. Por ejemplo, se podían desarrollar pruebas dentro de las horas de *Calidad y Auditoría*; o realizar controles de calidad en las horas de *Métodos de Desarrollo*. O incluso se podían combinar actividades de dos asignaturas distintas dentro de una misma hora. La idea era que los alumnos trabajasen tal como lo harían si estuviesen en una empresa real.

Cada alumno podía trabajar en sus tareas de manera individual, coordinándose en momentos puntuales con sus compañeros. Este factor es relevante, ya que permite a los alumnos trabajar de manera autónoma, utilizando las horas de aula para labores de coordinación, y evitando así tener que reunirse fuera del horario lectivo. Esta autonomía permitía la inclusión en los equipos de alumnos en régimen semi presencial. Estos alumnos trabajan normalmente a distancia, asistiendo sólo a determinadas actividades relevantes de grupo, como los *Sprint Planning Meeting*.

Para realizar las labores de coordinación diarias, Scrum recomienda realizar todos los días, a primera hora, una pequeña reunión denominada *Scrum Daily Meeting*. En ella, los miembros del equipo informan sobre el estado actual de sus tareas. Tras esta reunión, se resuelven los problemas que puedan haber surgido.

Finalmente, el último Jueves de cada sprint se llevaban a cabo dos actividades: la *Sprint Review* y la *Sprint Retrospective*. En la primera actividad, cada equipo mostraba al *Product Owner* la nueva versión del software producida, explicando y justificando las decisiones tomadas. El objetivo era que el *Product Owner* verificase que lo desarrollado fuese conforme a lo esperado, sugiriendo las modificaciones que estimase oportunas.

Por último, durante la *Sprint Retrospective*, cada equipo reflexionaba sobre qué había ido bien y qué había ido no tan bien durante el desarrollo de ese sprint. El objetivo era crear una lista de acciones que permitiesen potenciar lo positivo y reducir lo negativo.

4.3. Configuración del cuatrimestre

A la hora de distribuir los sprints dentro del cuatrimestre, la primera decisión que tomamos fue la de adelantar todo lo posible la ejecución del proyecto. Por ex-

periencia sabíamos que en el último tramo concurren dos factores que contribuyen negativamente al aprendizaje: (1) los alumnos comienzan a mostrar síntomas de fatiga; (2) la acumulación de entregas de diferentes prácticas y la cercanía de los exámenes les genera una cierta ansiedad que dificulta su concentración. Además, en el caso del primer cuatrimestre, la presencia tanto de las navidades como de los festivos previos impiden una ejecución sin interrupciones del proyecto.

No obstante, era necesario dedicar unas semanas iniciales a que los alumnos adquiriesen una mínima formación con la que poder afrontar el proyecto con ciertas garantías. Por otro lado, debía reservarse cierto tiempo en cada asignatura para el desarrollo de los objetivos de aprendizaje no integrados en el proyecto.

Por otro lado, experiencias previas nos aconsejaban realizar al menos tres sprints. En general, esperábamos que los resultados del primer sprint fuesen bastante pobres, ya que nadie nace sabiendo y lo natural, la primera vez que realiza algo, es equivocarse. En base a los comentarios de evaluación recibidos, se esperaba que los alumnos mejorasen sus habilidades durante el segundo sprint, alcanzando un nivel de madurez adecuado durante el tercero.

Teniendo en cuenta estos factores, las cuatro primeras semanas del cuatrimestre se dedicaron a la adquisición de unas bases teóricas mínimas para el desarrollo del proyecto. Además, cada equipo debía definir junto a su *Product Owner* el *Product Backlog* sobre el que luego trabajarían. A continuación, durante las semanas 5 a 10, se desarrollaban los tres sprints del proyecto integrado. El resto del cuatrimestre se dedicaba a trabajar los objetivos de aprendizaje no integrados en el proyecto. Con esta organización conseguimos que cada asignatura tuviese tiempo suficiente para desarrollar adecuadamente todos sus objetivos iniciales, no siendo necesario sacrificar ninguno de ellos.

4.4. Evaluación y calificación

Para facilitar la gestión administrativa de las asignaturas optamos por calificar sus actividades por separado. La correspondencia entre cada actividad y su asignatura era clara, así que cada actividad se evaluaba y calificaba dentro del contexto de esa asignatura. Las actividades que no correspondían a ninguna asignatura, tales como la implementación del producto, se repartían entre todas ellas. Por cada actividad se establecían unos criterios de evaluación únicos, por los que se reducía el riesgo de aparición de inconsistencias.

Este sistema nos permitía tratar con facilidad el caso de los alumnos no matriculados en alguna de las asignaturas del proyecto. En este caso, a cada alumno se le evaluaban sólo las actividades correspondientes a las asignaturas en las que estuviese matriculado, aunque era libre de participar en las actividades que quisiese.

- P1 ¿Consideras que realizar un mismo proyecto común te ayuda a ser más productivo? (1 - No; 5 - Sí)
 P2 Indique si considera los proyectos de cursos anteriores como realistas. (1 - No; 5 - Sí)
 P3 Indique si considera el proyecto desarrollado como realista. (1 - No; 5 - Sí)
 P4 ¿Has tenido la sensación durante el proyecto de estar trabajando para una empresa? (1 - No; 5 - Sí)
 P5 ¿Cómo ha sido tu motivación en comparación con la de proyectos anteriores? (1 - Menor; 5 - Mayor).
 P6 ¿El proyecto te ha ayudado a entender mejor cómo se integran los contenidos de cada asignatura? (1 - No; 5 - Sí)
 P7 ¿Cómo valoras que las asignaturas fuesen temporalmente cuasi-indistinguibles?. (1 - Muy Negativo; 5 - Muy Positivo)
 P8 ¿Cómo valoras que el proyecto fuese una aplicación Android? (1 - Muy Negativo.; 5 - Muy Positivo)
 P9 ¿Cómo valoras que se haya trabajado con un servicio de datos externo y real? (1 - Muy Negativo.; 5 - Muy Positivo)

Cuadro 3: Cuestionario de Evaluación.

| | 1 | 2 | 3 | 4 | 5 | avg |
|------------|---|----|---|---|----|------|
| P01 | 0 | 0 | 3 | 7 | 3 | 4.00 |
| P02 | 3 | 10 | 0 | 0 | 0 | 1.77 |
| P03 | 0 | 0 | 3 | 8 | 2 | 3.92 |
| P04 | 0 | 1 | 5 | 6 | 1 | 3.54 |
| P05 | 0 | 0 | 1 | 7 | 5 | 4.31 |
| P06 | 0 | 1 | 4 | 6 | 2 | 3.69 |
| P07 | 0 | 0 | 2 | 5 | 6 | 4.31 |
| P08 | 0 | 0 | 0 | 3 | 10 | 4.77 |
| P09 | 0 | 0 | 0 | 3 | 10 | 4.77 |

Cuadro 4: Resultados del Cuestionario.

La calificación se realizaba de modo tanto conjunto como individual, manteniendo un equilibrio entre la *interdependencia positiva* y la *exigibilidad individual*. Las actividades que eran inherentemente de grupo, tales como los *Sprint Planning Meeting*, tenían una calificación común al grupo. Para aquellas actividades que se realizaban de manera individual, como el desarrollo de las pruebas, la evaluación y calificación de cada actividad se entregaba exclusivamente al responsable de dicha actividad.

Los alumnos matriculados en régimen semipresencial debían asistir al menos una vez a las actividades de grupo evaluables. Teniendo en cuenta que las fechas de esas actividades eran conocidas desde el comienzo del cuatrimestre, disponían tiempo suficiente para realizar aquellas gestiones que fuesen necesarias. Las actividades individuales las podían realizar sin problemas de manera autónoma desde casa.

Los alumnos podían, y a veces debían, realizar una actividad varias veces. De este modo, en cada ejecución, y utilizando las evaluaciones previamente recibidas, los alumnos podían corregir los errores cometidos con anterioridad e ir mejorando su calificación, de acuerdo con un modelo de aprendizaje evolutivo. En nuestro caso, sólo se computaba la calificación de la última ejecución de cada actividad.

5. Evaluación y resultados

Los resultados obtenidos tras los dos años de ejecución del proyecto fueron bastante positivos. Desde el

punto de vista académico, todos los alumnos superaron las actividades propuestas y no se produjeron abandonos. Además, los comentarios recibidos fueron en su mayoría bastante buenos, y nuestras sensaciones como profesores muy positivas. Considerábamos que los alumnos habían aprendido más y mejor. En general, quedamos bastante satisfechos con la experiencia.

Tras el segundo año, y con objeto de evaluar más cuantitativamente nuestro proyecto, realizamos un cuestionario a los alumnos. Las preguntas de dicho cuestionario más relevantes para el propósito de este artículo se muestran en el Cuadro 3. Cada pregunta se respondía de acuerdo a una escala *likert*, cuyos valores se muestran al lado de cada pregunta. El cuestionario fue completado por 13 de los 15 alumnos matriculados en el proyecto. El Cuadro 4 muestra, para cada pregunta, el número de respuestas por cada valor de la escala *likert*, con el valor más frecuente destacado, y el valor medio de las respuestas.

Tal como se puede comprobar, los alumnos consideran en su mayoría que la integración de asignaturas les ayuda a ser más productivos (P01). Por tanto, nuestro problema *P1* (ver Apartado 2) se podía considerar como parcialmente satisfecho.

Además, la percepción de la veracidad del proyecto y la experiencia mejoró. Los alumnos consideraron el proyecto aproximadamente un 43 % más real que proyectos anteriores en los que habían trabajado (P02 y P03). Además, parte de estos alumnos sintieron la sensación de estar trabajando para una empresa durante el desarrollo del proyecto (P04), aunque en este caso, los resultados obtenidos nos indican que debemos seguir mejorando en este aspecto. Ambos factores tuvieron una influencia positiva sobre la motivación del alumno, que era bastante mayor que en experiencias parecidas (P05). Con estos resultados podíamos considerar nuestro problema *P2* como parcialmente resuelto.

Por último, los alumnos también consideraban que, en cierto modo, el proyecto les ayudaba a entender mejor cómo se integran y complementan los conocimientos de las tres asignaturas unificadas (P06). No obstante, los resultados indican que existe cierto margen de mejora, por lo que debemos seguir trabajando este

aspecto. Estos resultados indicaban que estábamos en el camino para resolver el problema P3.

Por otro lado, la valoración de otros aspectos del proyecto, como la rotura de barreras entre asignaturas (P07), o la utilización de Android (P08) y un servicio de datos externos (P09), también fue bastante positiva.

Merece también la pena destacar, aunque por razones de espacio dicho tema no se ha tratado en este trabajo, que el proyecto nos ha permitido desarrollar de manera muy adecuada ciertas competencias transversales como la *capacidad de trabajo en equipo* o la *capacidad de liderazgo*, obteniéndose muy buenos resultados en este ámbito.

Como aspectos a mejorar, los alumnos indicaron que se debería incrementar sustancialmente las horas que la asignatura de *Procesos de Ingeniería del Sw* dedica al proyecto integrado; ya que mientras que el proyecto requería del desarrollo de un alto volumen de pruebas; la asignatura encargada de dicha competencia sólo aportaba una hora al proyecto.

Por otro lado, nuestra propuesta podría tener problemas de escalabilidad. La asignación de un *Product Owner* distinto a cada grupo permite que actividades como las *Sprint Review* se puedan desarrollar de manera concurrente para todos los grupos, pero requiere de la implicación de un alto número de profesores. Este esquema podría no ser sostenible con grupos de más de 25 alumnos.

En segundo lugar, hemos recibido quejas de algún compañero, responsable de otras asignaturas, por la excesiva dedicación de los alumnos al proyecto. Este fenómeno de sobremotivación [6] puede y debe controlarse en nuestro caso, tal como hemos comentado, mediante la supervisión de la *velocidad de equipo*.

6. Sumario y trabajo futuro

Este trabajo ha presentado una experiencia de integración de tres asignaturas relacionadas con la Ingeniería del Software. Esta integración se realizó a través del desarrollo de un proyecto software común a las tres asignaturas. A diferencia de otras propuestas similares, la integración afectó sólo a una parte del cuatrimestre y a unos objetivos de aprendizaje muy concretos. Durante este tiempo, las asignaturas fueron prácticamente indistinguibles. Ello nos permitió reducir redundancias e inconsistencias, abordar proyectos de mayor tamaño y complejidad y trabajar mejor las conexiones entre asignaturas.

Animados por los buenos resultados obtenidos, en futuros cursos esperamos poder ampliar la integración a otras asignaturas de la intensificación, y, siguiendo las ideas de otros autores [3, 4], mejorar tanto el equipamiento de nuestras aulas para facilitar el traba-

jo cooperativo como el conjunto de herramientas y la infraestructura tecnológica utilizada para el desarrollo del proyecto.

Referencias

- [1] Cristina Barrado, Raúl Cuadrado, Luis Delgado, Fernando Mellibovsky, Enric Pastor, Marc Perez, Xavier Prats, Jose I. Rojas, Pablo Royo, y Miguel Valero. Una experiencia de unificación de asignaturas para desplegar PBL (y las quejas que originó). *ReVisión*, 6(2):50–59, Septiembre 2013.
- [2] Óscar Belmonte, Mercedes Segarra, Reyes Grangel, y Sergio Aguado. Desde la Iniciativa Empresarial hacia el éxito pasando por Metodologías Ágiles e Ingeniería del Software. En *Actas de las XXII Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, páginas 169–176, Almería, Julio 2016. Ponencia.
- [3] María José García García, Juan José Escribano Otero, y M^a Cruz Gaya López. Experiencia de aplicación de ABP al Grado de Ingeniería Informática. En *Actas de las XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, páginas 125–132, Oviedo, Julio 2014. Ponencia.
- [4] Daniel González-Peña, Miguel Reboiro Jato, y Francisco J. Ribadas-Pena. Entorno de integración continua para la docencia práctica de Java EE. En *Actas de las XXII Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, páginas 27–34, Almería, Julio 2016. Ponencia.
- [5] Thom Markham, John Larmer, y Jason Ravitz. *Project Based Learning*. Buck Inst for Education, Segunda Edición, Mayo 2003.
- [6] Joan Navarro, Xavi Canaleta, David Vernet, Xavi Solé, Virginia Jiménez, y Núria Costa. Motivación, desmotivación, sobremotivación, y daños colaterales. En *Actas de las XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, páginas 467–474, Oviedo, Julio 2014. Ponencia.
- [7] Jeff Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. Random House Business, 2015.
- [8] Miguel Valero y Javier García. Cómo empezar fácil con PBL. En *Actas de las XVII Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, páginas 109 – 116, Sevilla, Julio 2011.
- [9] Carlos J. Villagrà-Arnedo, Francisco J. Gallego-Durán, Rafael Molina Carmona, y Faraón Llorens-Largo. ABPgame+: siete asignaturas, un proyecto. En *Actas de las XX Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, páginas 285–292, Oviedo, Julio 2014.