

Evaluación de equipos de trabajo en el marco de Aprendizaje Basado en Proyectos y Métodos Ágiles

Patricio Letelier Torres
Departament de Sistemes Informàtics i
Computació
Universitat Politècnica de València
Valencia
letelier@dsic.upv.es

Héctor Cornide Reyes
Departamento de Ing. Informática y
Ciencias de la Computación
Universidad de Atacama
Copiapó-Chile
hector.cornide@uda.cl

Resumen

Existen muchos estudios que destacan las ventajas que tiene la aplicación de Aprendizaje Basado en Proyectos (ABP) en la enseñanza de Ingeniería de Software. Por otra parte, los Métodos Ágiles son la alternativa más apropiada para ámbitos de trabajo con alta incertidumbre, como lo es el desarrollo de software. Así, aplicar ABP basado en Métodos Ágiles es una combinación conveniente para docencia en Ingeniería de Software. Sin embargo, no existe consenso respecto del método más conveniente para la evaluación de los integrantes de un equipo de trabajo en el contexto de proyectos con Métodos Ágiles. En este trabajo se presenta un enfoque de evaluación aplicado en un marco de trabajo ABP en Ingeniería de Software para proyectos desarrollados usando Métodos Ágiles. Este enfoque permite obtener una evaluación para cada integrante de los equipos mediante tres mecanismos: evaluación a nivel de equipo (considerando el producto resultante y la adecuada aplicación de la metodología), coevaluación y autoevaluación (en estos dos últimos haciendo evaluación de Competencias Transversales). La evaluación de los estudiantes se integra de forma natural dentro del desarrollo proyecto, asociándose en cada Sprint a las Reuniones de Revisión y a las Reuniones de Retrospectiva. Además, el trabajo colaborativo y la gestión ágil del proyecto se apoyan en una herramienta que facilita la aplicación del enfoque de evaluación.

Abstract

There are many studies that highlight the advantages of the application of Project Based Learning (PBL) in the teaching of Software Engineering. On the other hand, Agile Methods are the most appropriate alternative for areas of work with high uncertainty, such as software development. Thus, applying PBL based on Agile Methods is a convenient combination for teach-

ing in Software Engineering. However, there is no consensus regarding the most convenient method for the evaluation of the members of a work team in the context of projects with Agile Methods. This paper presents an evaluation approach applied in an PBL framework in Software Engineering for projects developed using Agile Methods. This approach allows obtaining an evaluation for each member of the teams through three mechanisms: evaluation at the team level (considering the resulting product and the appropriate application of the methodology), co-evaluation and self-evaluation (in these both by doing evaluation of soft skills). The evaluation of the students is integrated naturally within the project development in Review Meetings and Retrospective Meetings carried on each Sprint. In addition, collaborative work and agile project management are supported by a tool that facilitates the application of the evaluation approach.

Palabras clave

Métodos de evaluación, Aprendizaje Basado en Proyectos (ABP), Métodos Ágiles, Competencias Transversales

1. Motivación

El Aprendizaje Basado en Proyectos [10] (ABP, y en inglés Project Based-Learning, PBL) acapara cada vez mayor atención y su aplicación se está extendiendo a todos los niveles de educación, desde primaria hasta universitaria. El “aprender haciendo” es la esencia del ABP y está alineada con las actuales y futuras exigencias del mundo laboral en cuanto a enfrentar retos técnicos y trabajar en entornos de trabajo colaborativo. ABP permite adquirir y aplicar conocimientos, pero también ofrece un escenario ideal para el desarrollo de Competencias Transversales (CT), las llamadas *soft skills* [12]. Para las asignaturas del área de la Ingeniería de Software, tanto

SWEBOK [2] como ACM/IEEE Curricula Guidelines [1] recomiendan incluir en los planes de estudios de las carreras la realización de proyectos en equipo para el desarrollo de dichas habilidades.

Para la gestión de proyectos (y gestión del trabajo en general), los Métodos Ágiles son el enfoque más apropiado en ámbitos con alta incertidumbre, donde se asume que ocurrirán cambios u otros imprevistos. Por esto, era de esperar que los Métodos Ágiles fueran adoptados rápidamente por la industria del software. Por su transversalidad, los Métodos Ágiles también se están consolidando en muchas otras áreas no relacionadas con desarrollo de software. Entre los métodos más populares están: Scrum, Kanban, Extreme Programming y Lean Development, siendo el primero el que más empresas declaran estar utilizando [9]. Así pues, parece razonable que tratándose de ABP aplicado en un contexto de Ingeniería de Software el trabajo se gestione usando Métodos Ágiles.

Cuando se aplica ABP surgen importantes desafíos y cuestionamientos respecto del método de evaluación de los estudiantes. La calificación final debe estar individualizada para cada estudiante, pero al intentar medir el aprendizaje y desempeño de cada uno de ellos en el proyecto se debe evitar afectar la autogestión del equipo, uno de los aspectos clave del enfoque ágil. Por lo anterior, es deseable que el propio desarrollo del proyecto integre de forma natural la evaluación de los integrantes del equipo. Si queremos recrear un proyecto con el mayor realismo posible no es conveniente añadir otros mecanismos de evaluación fuera del contexto del trabajo asociado al proyecto (por ejemplo, exámenes) y así procurar que el equipo mantenga siempre en el foco en el objetivo del trabajo con altos niveles de colaboración entre sus integrantes.

A continuación, comentamos algunos de los pocos trabajos que hemos encontrado respecto de mecanismos de evaluación en ABP y aplicando Métodos Ágiles. En el trabajo presentado por Goñi et al. [4] el proceso de evaluación no está integrado en el proceso de desarrollo del proyecto, la evaluación se basa en exposiciones al terminar cada Sprint. Cada equipo debe defender su trabajo frente a otro equipo, el cual debe elaborar un *informe de inspección* que es entregado al profesor. En otras experiencias de ABP realizadas en Ingeniería del Software [3, 8, 11], tampoco es posible observar un método de evaluación integrado en el proyecto, sino que los profesores van adecuando la evaluación según los objetivos que plantean al definir el proyecto ABP, y por esto no resulta extraño encontrar distintos instrumentos de evaluación en cada caso. Las rúbricas aparecen como el instrumento de evaluación más frecuente, seguida de la elaboración de portafolios del proyecto y la evaluación entre pares.

El objetivo de este artículo es presentar un marco de trabajo que combina ABP y Métodos Ágiles, que ofrece un enfoque para la evaluación de los estudiantes integrado de forma natural en el desarrollo del proyecto. Nuestra propuesta permite obtener una evaluación para cada integrante del equipo mediante tres mecanismos: evaluación del equipo (heteroevaluación asociada al resultado obtenido y su desempeño metodológico), coevaluación y autoevaluación, estos últimos desde una perspectiva de evaluación de Competencias Transversales.

Una característica importante de nuestra propuesta, es la baja carga en tiempo que tiene para el profesor realizar el proceso de evaluación y de retroalimentación a los equipos. Para realizar todo lo anterior, se usa un baremo totalmente alineado con la información proporcionada por una herramienta de apoyo para la gestión ágil del proyecto. Asimismo, la evaluación de Competencias Transversales está apoyada por rúbricas donde los propios estudiantes son los que la realizan mediante autoevaluación y coevaluación.

El artículo está dividido en 5 secciones. En la sección 2 se presenta la asignatura Proyecto de Ingeniería de Software, donde hemos aplicado nuestro enfoque. En la sección 3 se detalla el enfoque de evaluación propuesto. En la sección 4 se describe cómo el enfoque de evaluación se apoya en una herramienta para gestión ágil del proyecto. Finalmente, se presentan las conclusiones de este trabajo.

2. Asignatura Proyecto de Ingeniería de Software (PIN)

PIN es una asignatura de la Rama de Ingeniería de Software del Grado en Ingeniería Informática dictada en la Escuela Técnica de Ingeniería Informática (ETSIInf) de la Universitat Politècnica de València (UPV). PIN se imparte en el primer cuatrimestre del cuarto curso (cuatrimestre 4A) y tiene asignados 6 créditos ECTS, con una docencia presencial de 4.5 horas semanales, en una planificación de 14 semanas. El objetivo de PIN es que los estudiantes integren conocimientos técnicos y desarrollen habilidades transversales participando en un proyecto de desarrollo de software, en un contexto lo más cercano a la realidad industrial. Además, PIN aporta conocimientos metodológicos para la gestión de proyectos, particularmente aplicando Métodos Ágiles.

PIN es cursada por alrededor de 80 estudiantes cada año, organizados en dos clases, con lo cual se establecen 5 equipos de trabajo en cada clase (de entre 7-8 estudiantes cada uno). Cada equipo debe desarrollar una idea de negocio basada en un producto software, abordando desde la generación de la idea hasta la obtención de una primera entrega, que coincide con la presentación del producto en la Feria de Proyectos de Estudiantes de la ETSInf. Cada equipo

decide el producto que quiere desarrollar o elige entre propuestas ofertadas. Así también, elige las tecnologías y herramientas que utilizará para el desarrollo de su producto, con la excepción de la herramienta de apoyo al trabajo colaborativo y a la gestión ágil del proyecto, la cual es común para toda la asignatura. Esta herramienta se llama Worki¹. Los estudiantes de PIN, en su gran mayoría han cursado una asignatura previa en el segundo cuatrimestre de tercer curso llamada Proceso del Software (PSW), donde han sido entrenados en el uso de la herramienta Worki y han aprendido y ejercitado prácticas de Métodos Ágiles. Así, en PIN se asume dicha formación previa y el trabajo se enfoca por completo al desarrollo del producto software.

En PIN se aplica una metodología ágil basada en una combinación de prácticas ágiles del catálogo AgileRoadmap², que contiene un conjunto de prácticas extraídas de los Métodos Ágiles más populares (Scrum, Kanban, Lean Development y Extreme Programming). Así, y tal como se sugiere en AgileRoadmap, más que aplicar un Método Ágil específico la estrategia de PIN es aplicar prácticas ágiles, independientemente del método que las propone.

La Figura 1 ilustra el esquema de los proyectos en PIN. El proyecto tiene una fase de Puesta en Marcha (o Sprint 0) de unas 3 semanas en las cuales los equipos deben generar y evaluar su idea de negocio, esencialmente deben elaborar un Lean Canvas para sintetizar su idea de negocio, hacer un estudio comparativo con productos competidores y realizar una proyección muy simplificada de ingresos y gastos asociados a lo que sería la empresa basada en el producto que desarrolla cada equipo. Además, en la Puesta en Marcha el equipo ayuda al Product Owner (PO) a establecer el Backlog y a estimar de forma preliminar su contenido para así gestionar el alcance del proyecto. Finalmente, en la Puesta en Marcha el equipo también debe aprovechar de formarse en las tecnologías seleccionadas para el desarrollo. La fase de desarrollo está compuesta por 3 Sprints consecutivos, de alrededor de 3 semanas cada uno. Antes de comenzar cada Sprint se realiza la Reunión de Planificación del Sprint, y al finalizar el Sprint se llevan a cabo la Reunión de Revisión y la Reunión de Retrospectiva. Finalmente, la entrega del producto se realiza en la Feria de Proyectos, para la cual cada equipo debe elaborar un póster y preparar una demo para explicar su producto a los asistentes.

Durante el proyecto, el profesor desempeña los roles de PO y Scrum Master (SM). Como PO tiene la última palabra respecto de qué se debe implementar y en qué orden. Como SM observa y vela por la adecuada aplicación de las prácticas ágiles, apoyando al equipo en su mejora continua. Para dar más realismo

al proyecto, el PO intenta añadir nuevos requisitos durante el proyecto y, particularmente en las revisiones, presiona para incluir mejoras o evidenciar fallos de interpretación en los requisitos. De esta forma, constantemente se fuerza a que los equipos tengan que negociar el alcance del proyecto con el PO, quitando ítems del proyecto o estableciendo versiones reducidas de algunos ítems de gran tamaño (Épicas). En PIN solo se permite negociar alcance, no se negocian plazos ni costos del proyecto, debido a que queremos llevar a todos los equipos sincronizados en las fechas de Sprint y mantener la composición de integrantes de cada equipo. En las negociaciones los equipos deben basar su argumentación en su Capacidad (medida en Horas Ideales de Programación). Al comenzar el proyecto se considera una Capacidad teórica (antes de contar con datos del primer Sprint) de unas 5 Horas Ideales de Programación por cada integrante del equipo.

En cuanto a infraestructura, PIN dispone de un aula especialmente dedicada a ABP, donde se mantiene una distribución de mesas en islas para cada equipo, con suficientes tomas eléctricas como para conectar portátiles y un proyector.

3. Evaluación en PIN

En PIN se realizan tres tipos de evaluación: “Evaluaciones de equipo”, “Evaluaciones de CT” y “Autoevaluación final”.

Las “Evaluaciones de equipo”, corresponden a los hitos del proyecto e incluyen los siguientes 5 actos de evaluación (con sus pesos): Puesta en Marcha o Sprint0 (15%), Sprint1 (20%), Sprint2 (20%), Sprint3 (20%) y Entrega-Feria (15%). Estas evaluaciones usan dos mecanismos; el profesor pone una nota al equipo, y los integrantes del equipo, mediante coevaluación de la contribución de cada uno de sus integrantes, pueden retocar la nota del equipo para asignar una nota específica a cada integrante.

En las “Evaluaciones de CT”, se realiza autoevaluación-coevaluación de Competencias Transversales (CT), la cual tiene un peso de 5 % y se lleva a cabo al final de los Sprints 1, 2 y 3.

La “Autoevaluación final”, es una autoevaluación con un peso de 5%, la cual aborda la medición del aprendizaje de los principales conceptos metodológicos y situaciones enfrentadas durante el proyecto (se realiza la última clase de la asignatura).

Las “Evaluaciones de equipo” y “Evaluaciones de CT” se gestionan en un fichero de notas (un documento de hojas de cálculo). A la herramienta Worki se sube este fichero de notas para cada equipo, quedando disponible para consulta y edición durante todo el proyecto. Así, este fichero es editado por el profesor en su rol de PO y SM, y por los estudiantes cuando se coevalúan en las Reuniones de Retrospectiva.

¹ www.tuneupprocess.com

² agile-roadmap.tuneupprocess.com

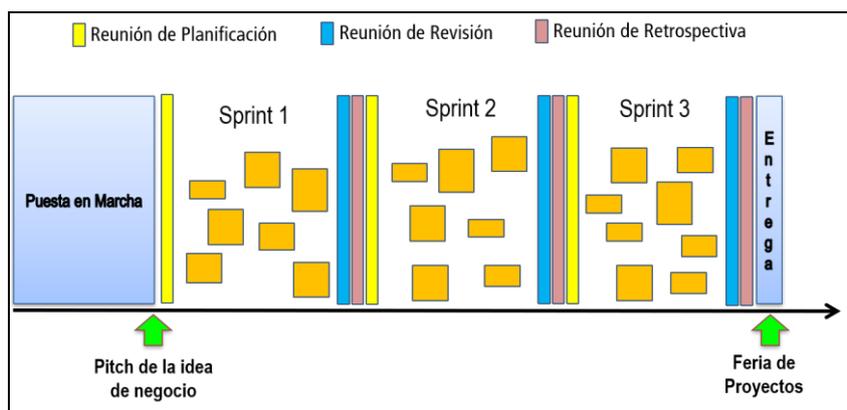


Figura 1. Esquema global de los proyectos en PIN

El fichero tiene celdas protegidas con contraseña evitando una edición indiscriminada, pero por otra parte la herramienta ofrece gestión de las versiones para los ficheros, con lo cual no hay mayor problema en dejar este fichero³ disponible para los estudiantes. A continuación, se describen con mayor detalle cada uno de los componentes de la evaluación en PIN.

3.1. Evaluaciones de equipo

Para cada uno de los actos de evaluación en el proyecto se ha establecido un baremo de evaluación del rendimiento del equipo. En el fichero de notas hay una hoja para cada acto de evaluación con el baremo correspondiente. En la Figura 2 se muestra la hoja del Sprint 2.

Como se observa en la tabla superior de la Figura 2, los criterios establecidos tienen diferentes pesos y el profesor asigna una calificación a cada criterio, lo cual genera una evaluación a nivel de equipo. Además, en las celdas de evaluación de un criterio el profesor puede introducir comentarios que estarán disponibles para consulta de los estudiantes.

Criterio	Peso (%)	Evaluación	Puntaje
Gestión del Backlog y Estructura del producto	5	9	0,5
Registros de tiempo invertido	10	7	0,7
Progreso del trabajo (Gráfica Burndown)	15	8	1,2
Terminación anticipada (Gráfica Terminadas/No Terminadas)	10	8	0,8
Ritmo de trabajo (Diagrama de Flujo Acumulado)	10	8	0,8
Gestión de pruebas de aceptación	20	6	1,2
Preparación del siguiente Sprint	15	9	1,4
Resultado del Sprint (y posible negociación al respecto)	15	9	1,4
	100		
		Nota equipo	7,9

Integrante	Contribución (%)
Integrante 1	10
Integrante 2	15
Integrante 3	15
Integrante 4	15
Integrante 5	10
Integrante 6	15
Integrante 7	15
Integrante 8	5
	100

Figura 2. Hoja de evaluación del Sprint 2

Los baremos para evaluación de Sprints son muy similares, hay ligeras variaciones de peso de los

criterios según el foco de mejora que debería tener cada Sprint. En el Sprint 3 no se incluye el criterio asociado a la preparación del próximo Sprint por ser el último del proyecto.

El profesor, como PO participa en la Reunión de Revisión del Sprint donde el equipo le enseña el resultado obtenido. Su apreciación como PO se verá reflejada en el criterio “Resultado del Sprint (y posible negociación al respecto)”. Luego, con el apoyo de la herramienta Worki, y desde una perspectiva de SM evalúa el desempeño del equipo respecto de la metodología ágil que se está aplicando. Esta evaluación provee información útil para el equipo en su Reunión de Retrospectiva.

La evaluación del equipo se complementa con la coevaluación que realizan los integrantes del equipo en el marco de la Reunión de Retrospectiva. Como se observa en la tabla inferior de la Figura 2, cada integrante es coevaluado respecto de su contribución⁴ (en este caso en el trabajo del Sprint 2). La suma de las contribuciones debe sumar 100%. De esta forma el equipo puede otorgar un reconocimiento (o penalización) a algunos integrantes, o simplemente, tener la misma contribución. Así, el porcentaje de contribución de cada integrante se aplica sobre la nota del equipo para obtener la evaluación individual. Si a un integrante se le otorga un porcentaje de contribución mayor a lo que sería la media del equipo le repercutirá como máximo en 1 punto adicional sobre la nota asignada al equipo. Por el contrario, una contribución bajo la contribución media del equipo provocará que la calificación del integrante sea menor que la nota asignada al equipo (sin límite hacia abajo, es decir, en el caso extremo de tener 0% de contribución la calificación sería un 0). Todos estos cálculos están incluidos como fórmulas en el fichero de notas.

Según lo anterior, en cada hito del proyecto, y en el marco de las Reuniones de Revisión del Sprint y de

³ <https://drive.google.com/file/d/1KTAhYJZwuTZSL8QVLb1FcwrtjW3gyr3/view>

⁴ Por “contribución” nos referimos a la aportación al resultado del Sprint, es decir, no se está midiendo la participación en términos de esfuerzo-tiempo dedicado, sino en eficacia.

Retrospectiva, cada miembro del equipo obtiene una calificación. Esto se refleja en la Hoja Resumen del fichero de notas, tal como se muestra en la Figura 3.

Integrante	Puesta en Marcha (10%)	Sprint 1 (15%)	Sprint 2 (20%)	Sprint 3 (25%)	Entrega (20%)	Final Proyecto
Integrante 1	6,4	8,5	6,3	9,1	7,1	7,6
Integrante 2	8,9	8,6	8,9	10,0	9,9	9,4
Integrante 3	8,9	6,6	8,9	8,5	9,9	8,6
Integrante 4	8,9	6,0	8,9	8,1	9,9	8,4
Integrante 5	8,9	8,6	6,3	9,2	7,1	8,0
Integrante 6	8,9	7,9	8,9	9,2	7,1	8,4
Integrante 7	6,4	6,6	8,9	9,2	3,6	7,1
Integrante 8	6,4	6,0	3,1	9,2	7,1	6,6
Nota Equipo	8,0	7,6	7,9	9,1	8,9	

Figura 3. Hoja resumen de notas de equipo

3.2. Evaluaciones de CT

La asignatura PIN evalúa 4 Competencias Transversales de entre las 13 que promueve la UPV⁵ (aunque el contexto de proyecto y métodos ágiles también es favorable para el desarrollo de otras CT, tal como se explica en [5]). Para la evaluación de CT en PIN hemos elaborado rúbricas que miden cuatro aspectos en cada CT. A continuación, se indican las CT evaluadas por PIN:

- CT04 - Innovación, creatividad y emprendimiento⁶: Innovación y creatividad; Investigación e integración de conocimientos; Motivación y resiliencia; Método
- CT05 - Diseño y proyecto⁷: Visión del producto/servicio; Elementos del proyecto; Estrategia de proyecto; Progreso del proyecto y riesgos
- CT06 - Trabajo en equipo y liderazgo⁸: Colaboración; Comunicación; Liderazgo; Compromiso
- CT12 - Planificación y gestión del tiempo⁹: Eficacia y puntualidad; Disciplina y organización; Gestión de compromisos; Planificación

Con el enlace en cada nombre de CT se puede acceder a una tabla con el detalle de la correspondiente rúbrica. En la Figura 4 se muestra la rúbrica para la CT Planificación y gestión del tiempo.

El fichero de notas contiene una hoja en la cual cada equipo debe completar las CT de cada integrante. Esto se realiza con refinamiento sucesivo en cada Reunión de Retrospectiva. Después de la Puesta en Marcha, cada integrante hace una autoevaluación inicial de sus CT. Posteriormente, al hacer cada Reunión de Retrospectiva, el equipo debe coevaluar a sus integrantes, modificando alguna evaluación de CT cuando lo considere necesario. Es decir, la evaluación de un integrante se inicia con su autoevaluación de CT, y según lo que perciban sus compañeros a lo largo del proyecto dicha evaluación puede mejorar o empeorar, siendo la última la que se traslada a la nota final. La idea es que en los casos que corresponda se genere en el equipo un diálogo constructivo respecto

a los aspectos en los cuales un integrante debe mejorar. Esta coevaluación de CT se hace al final de la Reunión de Retrospectiva.

CT12 – Planificación y gestión del tiempo				
	No Alcanzado [0-5]	En desarrollo [5-7]	Adecuado [7-9]	Excelente [9-10]
Eficacia y puntualidad	No cumple con sus plazos y/o a resultado esperado.	Cumple con los plazos de sus tareas pero no las sueltas entregar totalmente completadas o no tienen la calidad esperada.	Cumple con los plazos y resultado esperado de sus tareas.	Además de cumplir con sus tareas y correspondientes plazos se concentra en conseguir una solución satisfactoria para el cliente, evitando invertir esfuerzo en aspectos prescindibles y/o postergables.
Disciplina y organización	Muestra desorganización en su trabajo. No respeta las prioridades o no se centra en terminar tareas.	Se interesa por la organización del trabajo pero no de forma continuada. Suele distraerse con tareas no prioritarias.	Muestra un continuo interés por mantener organizado el trabajo. Se centra en terminar el trabajo que es prioritario.	Promueve mejoras en la organización del trabajo y contribuye al establecimiento de prácticas y métodos de consenso para abordar el trabajo.
Gestión de compromisos	No presta atención a la factibilidad de cumplir con los compromisos que se plantean. No se interesa en la estimación de las tareas ni si existe coherencia respecto de la Capacidad suya o del equipo para realizarlas.	Se interesa en estimar sus tareas pero no realiza reestimaciones oportunas ni se interesa por tener establecida su Capacidad o la del equipo para establecer compromisos.	Se interesa en realizar buenas estimaciones y en mantenerlas actualizadas. Se interesa en conocer la Capacidad suya y del equipo para establecer compromisos factibles de cumplir.	Evalúa el cumplimiento de los compromisos y los cálculos de estimación y Capacidad que se realizaron. Promueve la mejora de los mecanismos para estimación y cálculo de la Capacidad del equipo.
Planificación	No muestra interés en participar en la planificación del trabajo. Espera a que le indiquen las tareas que debe realizar.	Se interesa por la priorización y planificación del trabajo pero no participa activamente en su establecimiento.	Participa en la priorización y planificación del trabajo. Negocia con el cliente la planificación del trabajo para intentar garantizar su cumplimiento.	Evalúa el cumplimiento de la planificación y sugiere mejoras para que lo planificado coincida en mayor medida con lo real.

Figura 4. Detalle de la rúbrica para la CT12

3.3. Autoevaluación final

La autoevaluación se realiza mediante un formulario de Google Forms. El cuestionario consta de 20 afirmaciones para las cuales el estudiante debe responder con Sí o No. La evaluación equivale simplemente al porcentaje de respuestas “Si” sobre el total de afirmaciones. A continuación, se presentan dichas afirmaciones:

- Puedo explicar la diferencia entre versión y sprint.
- Puedo explicar la diferencia entre entrega y versión.
- Puedo explicar qué es un Mínimo Producto Viable (MVP).
- Puedo explicar la diferencia entre entrega y proyecto.
- Conozco al menos 3 formas como poder enfrentar el desarrollo de una épica y cuál es la más recomendada.
- Puedo explicar las diferencias y similitudes entre Historia de usuario y Caso de Uso.
- Conozco las responsabilidades de PO y SM.
- Conozco las 4 reuniones de Scrum.
- Sé participar eficazmente en una Retrospectiva.
- Soy capaz de explicar el concepto Fail Fast.
- Soy capaz interpretar correctamente una gráfica de flujo acumulado.
- Soy capaz de interpretar correctamente la gráfica Burn-down.
- Sé participar eficazmente en una Daily Meeting.
- Entiendo qué son las Pruebas de Regresión y por qué es importante aplicarlas.
- Entiendo por qué es interesante disponer de pruebas automatizadas.
- Soy capaz de explicar el concepto retrabajo.

⁵ <http://www.upv.es/contenidos/COMPTRAN/>

⁶ https://drive.google.com/file/d/1j8kxZ8S7CtD7N4qW0chMphix64q_A28r/view

⁷ <https://drive.google.com/file/d/1n3dtOK8gW1NOthe7EMftcq27WaUwyCau/view>

⁸ https://drive.google.com/file/d/1tiu3EhZl_Umohd_SYDlmm-BsD5FDBT7q/view

⁹ <https://drive.google.com/file/d/1rBLjaAAKV3qhdrrkK5bbJNCcJwnEKvu/view>

- Soy capaz de diseñar un tablero Kanban que refleje las actividades/estados de un proceso.
- Sé cómo trabajar con un tablero Kanban y Sprints.
- Sé cuándo se debe negociar. Sé calcular la Capacidad del equipo, en Horas Ideales o Puntos.
- Sé hacer estimaciones en Horas Ideales y entiendo su diferencia respecto de horas de contrato.

4. Evaluación apoyada por Worki

Uno de los aspectos clave en el método de evaluación en este contexto de ABP y Métodos Ágiles es que los estudiantes cuenten oportunamente con su evaluación durante el desarrollo del proyecto. Normalmente este requisito estaría en conflicto con hacer una evaluación detallada, dado que en nuestro caso tenemos que evaluar a dos clases, de 5 equipos de 7-8 estudiantes cada uno. Pero, como explicamos antes, parte de la evaluación se deja en manos de los estudiantes mediante coevaluación y autoevaluación. En cuanto a la evaluación realizada por el profesor, en esta sección veremos cómo los criterios de los baremos de cada hito pueden ser fácilmente evaluados con el apoyo de Worki. En Worki existen formularios específicos que, además de servir para realizar el seguimiento del proyecto, reflejan el desempeño metodológico del equipo. En particular, Worki ofrece un Dashboard de Sprint (y uno equivalente para el Proyecto) que incluye un variado conjunto de métricas y gráficas. Tomando como ejemplo el baremo del Sprint 2 (Figura 2), a continuación, se describe la evaluación de algunos de sus criterios (por limitaciones de espacio no podemos comentarlos todos).

4.1. Gestión del Backlog y Estructura del producto

Para este criterio consultamos la lista de UT¹⁰ que están en el Backlog (ver Figura 5), verificando que estén allí todas las UT que aún no se han incluido en un Sprint. En el Backlog, las UT deberían estar ordenadas y tener la información mínima requerida. Además, en todo momento en el Backlog tiene que estar delimitado el alcance del proyecto, es decir, deben estar señaladas cuáles UT se consideran dentro del alcance del proyecto y cuáles no. Para esto existe en las UT un campo llamado Proyecto, que en caso de tener el valor implica que está dentro del alcance del proyecto.

La Estructura del producto se representa con un árbol de nodos (Figura 6) que refleja la jerarquía de funcionalidades del producto. Cada UT está asociada

(afecta) a una parte de dicha estructura. Se debe verificar que se mantenga una adecuada estructura y que todas las UT estén asociadas a una o varias partes de ella.

Orden	UT	Proyecto	Punt...	Importancia	Urgencia	Riesgo
10	2865 - Modificación bonar perfil	Entrega 1 TRANSPORTIFY	1	Alta	Media	Alto
20	2869 - Avisos	Entrega 1 TRANSPORTIFY	5	Media	Media	Medio
30	2933 - Juntar y organizar pruebas automatizadas	Entrega 1 TRANSPORTIFY	8	Alta	Alta	Alto
40	2842 - Cancelar incidencias	Entrega 1 TRANSPORTIFY	1	Baja	Baja	Medio
50	2944 - Mejora separación logica y persistencia	Entrega 1 TRANSPORTIFY	5	Media	Media	Alto
60	2864 - Foto de perfil de usuario	Entrega 1 TRANSPORTIFY	3	Baja	Media	Medio
70	2346 - Notificaciones	Entrega 1 TRANSPORTIFY	3	Media	Media	Bajo
80	2344 - Recomendaciones	Entrega 1 TRANSPORTIFY	8	Alta	Media	Bajo
90	2338 - Opiniones	Entrega 1 TRANSPORTIFY	3	Media	Media	Bajo
100	2341 - Pagos	Entrega 1 TRANSPORTIFY	0.5	Baja	Baja	Medio
110	2578 - Negociación Envío		5	Media	Muy Baja	Medio
120	2699 - Puntos intermedios en viaje		1	Media	Baja	Medio
130	2330 - Ayuda		5	Media	Muy Baja	Bajo
140	2352 - Código entrega		2	Baja	Media	Medio
150	2347 - Chat		1	Baja	Media	Bajo
160	2800 - Inicio de Sesión mediante RRSS		3	Media	Media	Bajo

Figura 5. Lista de UT en el Backlog

Lista de UT	Pruebas de Aceptación
2843 - Asignar precio a paquete	
2868 - Autocompletar ciudad en selectores de ciudades	
2813 - Bug paquetes duplicados en búsqueda	
2794 - Selector de ciudades	
2756 - Puntos transportify sin nombre no se muestran bien	
2750 - Sistema filtros búsqueda	
2752 - Corrección Bug Búsqueda Paquetes y Viajes	
2323 - Buscar paquete	

Figura 6. Estructura del producto y UT asociadas

4.2. Registros de tiempo invertido

Respecto del tiempo registrado, en el panel de métricas mostrado en la Figura 7, se puede conocer el Tiempo total registrado (que reflejará la Capacidad del equipo) y los tiempos registrados por cada integrante. Se espera que el registro de tiempos sea coherente con la capacidad establecida para el equipo. Además, los integrantes del equipo deberían tener un tiempo invertido similar. La gráfica de la Figura 8 muestra la distribución de tiempo registrado en cada actividad y UT. Con ella podemos verificar si las proporciones de tiempo invertido se ajustan a lo esperado. La gráfica de retrabajo mostrada en la Figura 9 permite observar cuando en una UT hay una proporción significativa de retrabajo.

¹⁰ En Worki una UT (Unidad de Trabajo) es un ítem de trabajo, es decir, equivale al concepto de Historia de Usuario. En el caso de desarrollo de software una UT puede ser un Nuevo requisito, una Mejora en un requisito ya implementado o una Corrección de un fallo. También una UT puede representar un trabajo no asociado a cambios en el producto, por ejemplo, montar un servidor o preparar una demo.



Figura 7. Métricas globales del Sprint

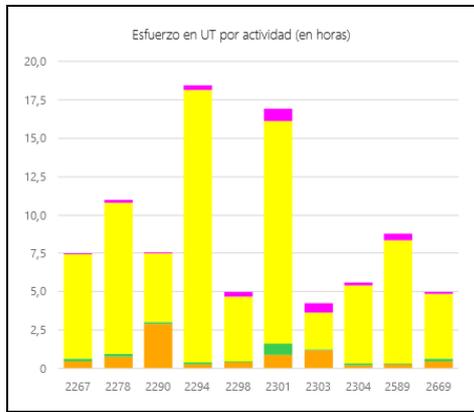


Figura 8. Tiempos en cada actividad y UT

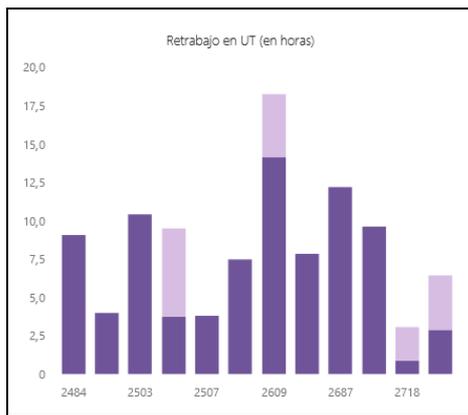


Figura 9. Retrabajo en cada UT

4.3. Progreso del trabajo

La gráfica Burndown (línea roja en la Figura 10) permite ver la tendencia del trabajo restante. Se espera que el tiempo restante, progresivamente, se vaya extinguiendo, es decir, que en la gráfica la línea roja se mantenga alineada con la línea de referencia que comienza en el esfuerzo total y termina en 0 al finalizar el Sprint. Correspondientemente, se espera que la línea azul (Gráfica Burn Up) aumente simétricamente respecto de la gráfica Burndown. La línea verde representa la estimación del Sprint y ayuda a visualizar cambios en las estimaciones (o en alcance). Además, debe verificarse que la gráfica Burndown no solo descienda porque se está invirtiendo tiempo (y así reduciendo el tiempo restante), sino que también se terminen trabajos proporcionalmente (UT terminadas). Esto se puede verificar con la gráfica de la Figura 11, donde la proporción verde es de UT terminada.

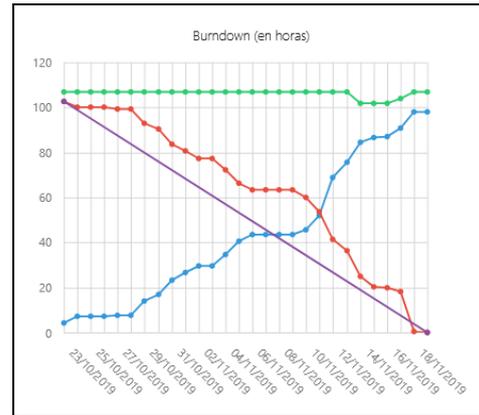


Figura 10. Gráficas Burndown y Burn Up

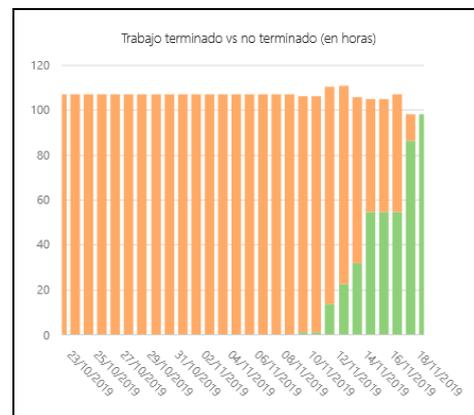


Figura 11. Proporción de trabajo terminado

4.4. Gestión de pruebas de aceptación

Para verificar la correcta gestión de pruebas de aceptación basta con acceder al árbol de nodos dentro de una UT y comprobar que se han definido y aplicado las correspondientes pruebas de aceptación. La Figura 12 muestra esta interfaz.

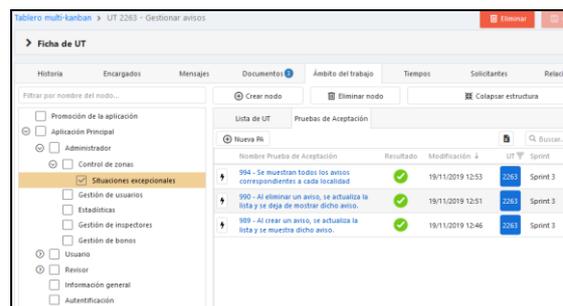


Figura 12. Gestión de pruebas de aceptación en UT

5. Conclusiones

Hemos presentado el enfoque utilizado para la evaluación de los estudiantes en el marco de trabajo de la asignatura PIN, donde se combinan ABP y Métodos Ágiles. El método docente y de evaluación de PIN se ha ido mejorando progresivamente desde hace más de

10 años de aplicación en docencia. En las Jenui del 2013 [3] y del 2015 [4] se presentaron principalmente los contenidos y organización de PIN, mientras que en este trabajo nos hemos centrado en el método de evaluación y su apoyo con una herramienta para la gestión ágil del proyecto. La evaluación del equipo se ha conseguido integrar de forma natural en el seguimiento del proyecto, enmarcándola en las Reuniones de Revisión y de Retrospectiva asociadas a cada Sprint. Después de realizar la evaluación de la Puesta en Marcha (Sprint 0) y del Sprint 1 los estudiantes ya logran asimilar estos hábitos de evaluación como parte de rutina del proyecto.

En los últimos años, hemos alineado el baremo de evaluación del equipo con el Dashboard que ofrece la herramienta Worki. Con esto, se ha conseguido simplificar la evaluación del equipo en cuanto a su desempeño metodológico y producto resultante. Así, en menos de 20 minutos el profesor puede hacer la evaluación del Sprint de un equipo.

Los estudiantes participan en la coevaluación de sus compañeros de equipo en dos oportunidades; al establecer las contribuciones de cada integrante y al evaluar las CT. Contando con criterios claros y transparentes (baremos, datos en la herramienta y rúbricas) no se presentan mayores inconvenientes en la coevaluación siendo bastante rápida de realizar. En la medida que hemos avanzado en estos aspectos, ha sido notable la desaparición de disconformidad que expresaban los estudiantes con su evaluación en años anteriores. Además, dado que se realizan 5 evaluaciones consecutivas siguiendo el mismo esquema, los estudiantes tienen la posibilidad de revertir oportunamente una situación negativa sin que su nota se vea demasiado afectada.

En las dos últimas ediciones de PIN se ha incluido la presentación de los productos resultantes en la Feria de Proyectos. Este aditivo ha dado un impulso de motivación y buen hacer al final del proyecto. Además, en la Feria se cuenta con la asistencia de empresas y profesores que le das mayor realce a las demos que ofrecían los equipos expositores.

Worki es una herramienta desarrollada en la UPV (que ofrecemos de forma gratuita para utilizar en docencia). Nos planteamos incluir este enfoque de evaluación dentro de sus funcionalidades, prescindiendo así del actual fichero de notas. Otros trabajos futuros están asociados utilizar las métricas y gráficas del Dashboard para medir el rendimiento del equipo, más allá de una evaluación docente, explorando cierta aplicación en contextos de equipos de trabajo en empresas.

Referencias

[1] Ardis, M., Budgen, D., Hislop, G. W., Offutt, J., Sebern, M., & Visser, W. (2015). SE 2014:

- Curriculum guidelines for undergraduate degree programs in software engineering. *Computer*, (11), 106-109.
- [2] Bourque, P., & Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press.
- [3] González, L. S., González, M. Á. C., Castro, R. F., & Cendón, J. A. Experiencia de aprendizaje basado en la implementación colaborativa de proyectos para el desarrollo de competencias emprendedoras. XVIII simposio internacional de informática educativa, SIIE 2016 (p. 109).
- [4] Goñi, A., Ibáñez, J., Iturrioz, J., & Vadillo, J. Á. (2014). Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software. En Actas de las XX Jenui, Oviedo, julio 2014.
- [5] Letelier, P. (2013). Competencias transversales basadas en prácticas ágiles. En Actas de 3rd International Conference on Innovation, Documentation and Teaching Technologies (INNO-DOCT 2015). Valencia, mayo 2013.
- [6] Letelier, P. y Penadés, M.C. (2013). Una estrategia para la enseñanza de metodologías ágiles. En Actas de las XIX Jenui, pp. 217 – 224, Castellón de la Plana, julio 2013.
- [7] Letelier, P. (2015). Una actividad para enseñar el uso de tableros kanban y diagramas de flujo acumulado. En Actas de las XXI Jenui, pp. 288 – 295, Andorra la Vella, julio 2015.
- [8] Martí, E., Poveda, F., Gurguí, A., & Gil, D. (2011). Aprendizaje Basado en Proyectos en Ingeniería Informática. Resultados y reflexiones de seis años de experiencia. En Actas de las XVII Jenui, Sevilla, julio 2011.
- [9] Raunak, M. S., & Binkley, D. (2017, September). Agile and other trends in software engineering. In 2017 IEEE 28th Annual Software Technology Conference (STC) (pp. 1-7). IEEE.
- [10] Savery, John R. (2006) "Overview of Problem-based Learning: Definitions and Distinctions". *Interdisciplinary Journal of Problembased Learning*: Vol. 1: Iss. 1, Article 3.
- [11] Vázquez-Ingelmo, A., García-Peñalvo, F. J., Rodríguez-Conde, M., & García-Holgado, A. (2019). Resultados preliminares tras tres años aplicando aprendizaje basado en proyectos en ingeniería del software (No. COMON-2019-CINAIC-0141).
- [12] Voogt, J., & Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *Journal of curriculum studies*, 44(3), 299-321.