

# Una estrategia para la enseñanza de metodologías ágiles

Patricio Letelier Torres

Depto. Sistemas informáticos y Computación  
Universidad Politécnica de Valencia  
letelier@dsic.upv.es

M<sup>a</sup> Carmen Penadés Gramaje

Depto. Sistemas informáticos y Computación  
Universidad Politécnica de Valencia  
mpenades@dsic.upv.es

## Resumen

El propósito de este trabajo es presentar una estrategia docente para la enseñanza de metodologías ágiles, que hemos refinado durante más de 10 años de aplicación en asignaturas de Ingeniería Informática, y que actualmente acabamos de trasladar al Grado en Ingeniería Informática. Se trata de 2 asignaturas consecutivas que integran tanto el aprendizaje de métodos y técnicas para el desarrollo de software como su puesta en acción en el marco de un proyecto de desarrollo de software. Si bien se ofrece una visión global de las metodologías, gran parte del contenido de estas asignaturas se centra en el enfoque ágil. En la asignatura *Proceso del Software* (PSW) se introducen los conceptos de Kanban [1], Lean Software Development [6], Scrum [3,7] y Extreme Programming [2], y se complementa su aprendizaje con el apoyo de diversos ejemplos y actividades. También en esta asignatura se realiza un trabajo práctico de exploración y planificación de un proyecto ágil. En la asignatura *Proyecto de Ingeniería de Software* (PIN) se forman equipos de 8 a 10 integrantes y se recrea el desarrollo ágil de un producto software empleando diversas prácticas ágiles, siguiendo un proceso iterativo e incremental, realizando 3 sprints, y todo ello apoyado con una herramienta para la gestión del proyecto. La organización y todo el material de PSW está disponible en [psw.tuneupprocess.com](http://psw.tuneupprocess.com).

## Abstract

The purpose of this paper is to present a teaching strategy that we have refined over 10 years of application in computer engineering courses, and now we have just moved to the new Degree in Computer Engineering. These two consecutive subjects integrate both, the learning of software development methods and practices and its implementation in the context of a project. We provide an overview of methods and practices, but certainly most of the contents of these subjects are focused on agile methods. The subject *Software Process* introduces the concepts and practices included in

the most popular agile methods; Kanban, Lean Software Development, Scrum and Extreme Programming, and their learning is complemented with a practical activities. In this subject the students also work on the exploring and planning phase of an agile project. In the subject *Software Engineering Project* working in teams participate in the agile development of a software product using a set agile practices and using a tool for collaboration support, following an iterative and incremental process, and carrying on three sprints. All the educational materials of PSW subject are available for download from [psw.tuneupprocess.com](http://psw.tuneupprocess.com).

## Palabras clave

Metodologías ágiles, prácticas ágiles.

## 1. Motivación

En el currículum de estudios de ingeniería del software los contenidos asociados a metodologías de desarrollo de software cumplen un papel fundamental. Si bien es imprescindible que el alumno se forme en técnicas asociadas a las diversas actividades del desarrollo (planificación, requisitos, diseño, programación y pruebas, entre otras), es también muy importante que sea capaz de poner todo ese conocimiento en práctica, de forma integrada en el contexto de un proyecto, y trabajando en equipo. Hasta principios de la década pasada, en el ámbito metodológico se enseñaba de forma exclusiva el enfoque tradicional, teniendo como representante más destacada a la metodología Rational Unified Process (RUP, producto de IBM) o bien su correspondiente propuesta contenida en el libro del Proceso Unificado [4]. Con menos popularidad, en el contexto de España, también se ha tenido como alternativa metodológica tradicional a Métrica [5].

En los últimos años las metodologías ágiles se han hecho un hueco en el ámbito industrial y representan una clara demanda del mercado en cuanto a la formación con la que deberían contar los titulados de informática. Sin embargo, en el contexto de las metodologías ágiles no existen referentes inte-

gradores sino más bien diversos métodos que coinciden en ciertos principios y prácticas, pero que no constituyen un cuerpo de conocimiento consensuado y reconocido. La explicación teórica e independiente de cada práctica o método (por ejemplo, las prácticas de Scrum o la explicación del método Kanban) resulta relativamente sencilla. El verdadero desafío es su aplicación eficaz en un contexto particular, dado por el equipo, dominio de aplicación, tecnología, cliente-contrato, etc. Las críticas dirigidas a las metodologías ágiles se refieren a los inconvenientes para ser introducidas en entornos que ya trabajan de una forma tradicional, o para escalarlas en un contexto dado que pueda requerir procesos más exigentes en cuanto a prácticas y técnicas. Como contrapartida, hay que reconocer que las metodologías tradicionales, aunque ofrecen un planteamiento más global y ambicioso de prácticas y técnicas, tienen serias dificultades para ser adaptadas y aplicadas eficientemente en el contexto de proyectos “pequeños”. Hasta ahora las comunidades tradicionales y ágiles se han mantenido enfrentadas y la mayoría de la información disponible refleja este hecho.

En el contexto de la Universidad Politécnica de Valencia, comenzamos a finales de los 90s instaurando la enseñanza de metodologías en asignaturas de último curso de Ingeniería Técnica Informática e Ingeniería Informática. En un comienzo nos centramos exclusivamente en metodologías tradicionales, luego fuimos incluyendo algunos contenidos de metodologías ágiles, hasta que en la actualidad, aunque mantenemos un enfoque global del proceso del software, nos hemos centrado en el enfoque ágil. Nuestra estrategia ha sido refinada a lo largo del tiempo y en su estado actual se basa en los siguientes aspectos:

- Si bien hacemos una introducción a los métodos ágiles más populares, lo que se enfatiza son las prácticas que hay detrás de cada método. Consideramos más importante el conocer el conjunto de prácticas ágiles, saber seleccionar las adecuadas al contexto y saber aplicarlas, que el centrarse en enseñar un método ágil específico.
- Damos mucha importancia a la estrategia de implantación de prácticas ágiles y los posibles obstáculos para dicha implantación, ofreciendo pautas al respecto.
- Si bien explicamos las diferencias del enfoque ágil respecto del tradicional, no insistimos en el típico enfrentamiento ágil-tradicional, incluso dejamos abierta la posibilidad de realizar mezclas de prácticas ágiles con algunas provenientes del enfoque tradicional.

- Utilizamos una herramienta para gestión ágil de equipos de trabajo, desarrollada por nosotros ([www.tuneupprocess.com](http://www.tuneupprocess.com)), y que está totalmente acoplada con nuestro planteamiento, aunque esto no impide que pueda utilizarse otras herramientas.

Es importante destacar que la mejora de contenidos y del enfoque docente a lo largo del tiempo se ha visto complementado con nuestra experiencia en numerosos proyectos reales y con asesorías en mejora de procesos que hemos realizado colaborando con equipos de trabajo en empresas.

El objetivo de este trabajo es describir nuestra estrategia para la enseñanza de metodologías ágiles. En concreto el plan de estudios vigente se compone de dos asignaturas: Proceso del Software (PSW) y Proyecto de Ingeniería del Software (PIN), que hemos coordinado estrechamente para ofrecer un visión global del proceso software, aunque centrándonos mayoritariamente en métodos ágiles. Además, hemos creado una página web ([psw.tuneupprocess.com](http://psw.tuneupprocess.com)) que muestra el programa detallado de PSW y contiene los enlaces para descargar todo el material utilizado, incluyendo presentaciones, ejemplos, actividades, preguntas de examen, plantillas para evaluación, etc.

La estructura del resto del artículo es la que sigue. En la sección 2 se presenta el método docente de ambas asignaturas. En la sección 3 las técnicas de evaluación utilizadas. En la Sección 4 las lecciones aprendidas a lo largo de los años de impartir metodologías ágiles. En la Sección 5 las herramientas de apoyo utilizadas. Finalmente se presentan las conclusiones, que incluyen una valoración de los alumnos respecto a la estrategia seguida.

## 2. Método docente

Proceso del Software (PSW) y Proyecto de Ingeniería de Software (PIN) son dos asignaturas del módulo de especialización en Ingeniería del Software dentro del Grado en Informática, impartidas en los cuatrimestres 3B y 4A, respectivamente. PSW cuenta con 4.5 créditos y PIN con 6 créditos. Para ambas asignaturas tenemos una programación cronológica de 14 semanas en las cuales se aborda trabajo de teoría, seminario y laboratorio (en el caso de PIN solo se trata de trabajo de seminario y laboratorio). Hemos concentrado en PSW la presentación y entrenamiento en conceptos y prácticas ágiles, dejando en PIN el protagonismo de su aplicación en el marco de la recreación de un proyecto de desarrollo.

Contamos con alrededor de 50 alumnos matriculados que cursan ambas asignaturas, pues habrán seleccionado el módulo de especialización en Ingeniería del Software. Los alumnos se distribuyen

en dos grupos, más o menos equilibrados en cuanto a número de alumnos.

## 2.1. Programa de PSW

La Figura 1 muestra el programa de la asignatura PSW (tal como de muestra en su página web). Contamos con alrededor de 14 semanas lectivas. Cada semana tenemos una sesión de teoría, y en la mayoría de ellas tenemos como complemento la presentación de ejemplos y la realización de actividades. Es decir, en el horario de aula se reserva un tiempo para comentar los ejemplos que ayudan a ilustrar los conceptos explicados en la correspondiente sesión de teoría, y también se realizan las actividades que permiten a los alumnos poner en acción conceptos o ejercitar prácticas ágiles. En la Figura 1 se remarcan los enlaces correspondientes para la descarga de las pautas y material utilizado en cada ejemplo y actividad.

En la parte práctica (sesiones en laboratorio) se realizan 10 sesiones, que empiezan en la tercera semana de la asignatura para dar tiempo a ver antes la introducción a las metodologías ágiles en la parte de teoría. El propósito de la parte práctica es que los alumnos elaboren una planificación ágil de un proyecto de desarrollo de software (propuesto por ellos mismos), realizando la exploración y captura ágil de requisitos, elaborando un plan iterativo de desarrollo y realizando la preparación de un sprint. Esta preparación del primer sprint incluye la selección de ítems candidatos (desde un Backlog priorizado) y para cada uno de ellos la elaboración de bocetos de interfaz, definición de pruebas de aceptación, y un diseño muy preliminar para poder realizar una estimación. Una vez preparados los ítems candidatos para el primer sprint, se confirma el sprint contrastando las estimaciones con la capacidad del equipo para el período del sprint. El profesor desempeña el rol de cliente e instructor para cada equipo. También en la parte práctica los alumnos aprenden a colaborar a través de una herramienta. El trabajo es realizado en equipos de 4 integrantes sin hacer distinción de roles entre ellos. En las 5 primeras sesiones de prácticas los equipos se entrenan tanto en la herramienta que utilizarán como en las tareas que realizarán, trabajando en un Caso de Estudio que ilustra todo el trabajo que posteriormente tendrán que desarrollar con su producto. El trabajo de prácticas concluye con la presentación que realiza cada equipo de su producto al final de la asignatura.

## 2.2. Programa de PIN

En PIN recreamos el trabajo de un equipo en el desarrollo de un producto software siguiendo un proceso incremental que mezcla principalmente prácticas de Kanban y Scrum. El trabajo realizado

en PIN es aplicado, aunque durante toda la asignatura el profesor destaca y refuerza los aspectos teóricos que se van poniendo en práctica. Los alumnos forman equipos de 8 a 10 integrantes. Al iniciar la asignatura los equipos seleccionan un producto entre los que fueron elaborados en la asignatura PSW, es decir, cada equipo comienza el desarrollo de uno de los productos preparados en la asignatura PSW. Cada equipo realiza tres sprints, y en el último sprint se debe conseguir lo que sería una entrega a producción de su producto. Las fechas de término de cada sprint son establecidas al comienzo de la asignatura cuidando que cada sprint tenga alrededor de 3 semanas efectivas de trabajo, es decir, sin considerar festivos ni vacaciones.

En los tres sprints realizados los equipos tienen la oportunidad de mejorar su forma de trabajo, experimentando con diferentes configuraciones de proceso e incorporando prácticas adicionales en cada sprint. Esta mejora continua de proceso es dirigida por el profesor, el cual enfatiza los elementos configurables y el proceso que está presente en cada sprint. Al final de cada sprint, inmediatamente después de la Reunión de Revisión, se realiza una Reunión de Retrospectiva para evaluar el desempeño del equipo y establecer mejoras para el sprint siguiente. La intervención del profesor durante toda la asignatura es muy importante en cuanto a ir enfatizando los errores que comete el equipo, y correspondientemente haciendo recomendaciones para que el equipo consiga mejorar su desempeño. Con nuestra experiencia hemos podido acotar y reproducir de forma bastante predecible los acontecimientos que se presentan durante el trabajo de cada equipo, y para ello tenemos definidas pautas para cada sprint. Estas pautas incluyen las recomendaciones a los equipos en cuanto a prácticas ágiles que se podrían aplicar o reforzar durante el proyecto. Algunos ejemplos de lo que se incluye en dichas pautas se indican a continuación: (a) “En el primer sprint los equipos trabajarán con pre-asignación total de las actividades a los miembros del equipo (lo cual generará ineficiencias en cuanto a esperas por trabajo que tiene asignado otro compañero)”; (b) “En el primer sprint se les debe insistir en la validación con el cliente, tanto de los bocetos como de las pruebas de aceptación”.; (c) “Penalizar al equipo si en la Reunión de Revisión el cliente descubre funcionalidad no obvia que no fue validada durante el sprint”; (d) “En el segundo sprint se les debe añadir como mejora el llevar un seguimiento más preciso del avance del sprint para así poder negociar oportunamente con el cliente”; (e) “En el segundo sprint presionar a los equipos para que incorporen funcionalidades no previstas y que excedan su capacidad en el sprint (obligándolos a negociar)”; etc.

Aula		Laboratorio	
	Presentación de la Asignatura ( <a href="#">Pautas Inicio de Prácticas</a> )		
1	<b>Unidad 1:</b> Introducción al Proceso del Software ( <a href="#">Presentación 1</a> ) ( <a href="#">Lecturas Complementarias</a> ) <b>Actividad:</b> Hard Choices ( <a href="#">Material</a> ) ( <a href="#">Pautas</a> )		1
2	<b>Unidad 2:</b> Metodologías Tradicionales ( <a href="#">Presentación 2</a> ) ( <a href="#">Lectura Complementaria</a> )		2
3	<b>Unidad 3:</b> Metodologías Ágiles - Introducción ( <a href="#">Presentación 3a</a> )	INICIO Caso de estudio de entrenamiento ( <a href="#">Indicaciones</a> ) ( <a href="#">Documento Visión</a> ) ( <a href="#">Unidades de Trabajo</a> ) ( <a href="#">Hoja Evaluación Prácticas</a> )	3
4	<b>Unidad 3.1:</b> Gestión Ágil de Requisitos ( <a href="#">Presentación 3.1</a> ) ( <a href="#">Lecturas Complementarias</a> ) <b>Ejemplo:</b> Documento Visión ( <a href="#">Documento</a> ) ( <a href="#">Pautas</a> ) <b>Ejemplo:</b> Pruebas de Aceptación ( <a href="#">Documentos</a> ) ( <a href="#">Pautas</a> ) <b>Actividad:</b> TDRE - Identificando PAs ( <a href="#">Pautas</a> )	<b>Actividad:</b> Planning Póker ( <a href="#">Material</a> ) ( <a href="#">Pautas</a> )	4
5	<b>Actividad:</b> Ilustrando el concepto WIP ( <a href="#">Pautas</a> ) <b>Ejemplo:</b> Cumulative Flow Diagram (CFD) ( <a href="#">Hoja</a> ) ( <a href="#">Pautas</a> )	<b>Actividad:</b> Herramienta soporte Kanban ( <a href="#">Pautas</a> )	5
6	<b>Unidad 3.2:</b> Prácticas y Métodos Ágiles ( <a href="#">Presentación 3.2</a> ) ( <a href="#">Lecturas Complementarias</a> )	<b>Actividad:</b> Ilustrando Kanban y CFD ( <a href="#">Hoja</a> ) ( <a href="#">Pautas</a> )	6
7	<b>Unidad 3.3:</b> Agilismo en acción ( <a href="#">Presentación 3.3</a> ) <b>Ejemplo:</b> Multi-kanban en Tune-up ( <a href="#">Pautas</a> )	ENTREGA Caso de estudio de entrenamiento	7
<b>Examen 1 (Ejemplo de Examen 1)</b>			
8	<b>Unidad 3.4:</b> Equipo Ágil ( <a href="#">Presentación 3.4</a> ) <b>Actividad:</b> Trabajo usando kanban físico ( <a href="#">Pautas</a> )	INICIO Caso de estudio Visión producto, Identificación Wus ( <a href="#">Pautas</a> ) <b>Reunión:</b> Brainstorming ( <a href="#">Pautas</a> )	8
9	<b>Actividad:</b> Uso de funcionalidades en producto ( <a href="#">Pautas</a> ) ( <a href="#">Hoja</a> ) <b>Ejemplo:</b> Gestión del Product Backlog ( <a href="#">Pautas</a> )	ENTREGA Visión Producto, Sprint 1 y Backlog	9
10	<b>Unidad 3.5:</b> Planificación y Seguimiento Ágil ( <a href="#">Presentación 3.5</a> ) <b>Actividad:</b> Sixes Game ( <a href="#">Hoja</a> ) ( <a href="#">Pautas</a> )	PREPARACIÓN Sprint 1	10
11	<b>Ejemplo:</b> Seguimiento de Sprints y Proyectos ( <a href="#">Pautas</a> )		11
12	<b>Unidad 3.6:</b> Implantación del Agilismo ( <a href="#">Presentación 3.6</a> ) <b>Actividad:</b> AGILE Roadmap ( <a href="#">Pautas</a> )	ENTREGA Sprint 1 preparado	12
13	<b>Unidad 3:</b> Metodologías Ágiles - Conclusiones ( <a href="#">Presentación 3b</a> ) <b>Actividad:</b> Ball Point Game ( <a href="#">Pautas</a> )	Co-evaluación de equipo por equipo (Pautas en Hoja de Evaluación Prácticas)	13
14	Conclusiones de la Asignatura ( <a href="#">Presentación</a> )		14
14	Presentaciones de Productos, su Sprint 1 y Backlog ( <a href="#">Pautas</a> ) Co-evaluación de presentación de equipo por participante ( <a href="#">Fichero</a> ) Auto-evaluación ( <a href="#">Fichero</a> ) Encuesta ( <a href="#">Preguntas</a> )		14
<b>Examen 2 (Ejemplo de Examen 2)</b>			

Figura 1: Programa de la asignatura PSW (tal como se muestra en su página web psw.tuneupprocess.com)

### 3. Método de evaluación

En ambas asignaturas realizamos evaluación continua. En PSW tenemos 7 actos de evaluación, los cuales se describen a continuación, indicando sus respectivos pesos en la nota final:

- 2 exámenes de respuesta abierta (40%). En cada uno se hacen alrededor de 10 preguntas de respuesta muy breve. Estos exámenes duran 45 minutos y se corrigen inmediatamente después de que los alumnos entregan el examen. Los exámenes son corregidos de forma cruzada por los alumnos y siguiendo las pautas del profesor, lo cual genera una interesante discusión respecto de cada pregunta.
- 4 entregas asociadas a las prácticas (55%): (a) práctica de entrenamiento en la herramienta de apoyo y en el proceso que se aplicará (15%), (b) documento visión del producto e identificación global de requisitos (15%), (c) preparación del primer sprint (15%) y (d) presentación del producto con su primer sprint y Backlog (10%). Para las entregas (c) y (d) se realiza co-evaluación; en la entrega (c), siguiendo pautas dadas por el profesor, los equipos se evalúan entre ellos, y en la presentación (d), y también siguiendo pautas proporcionadas por el profesor, cada equipo evalúa la presentación de producto de los otros equipos.
- 1 auto-evaluación que cada alumno hace de su desempeño y aprendizaje en la asignatura (5%). Para esto a los alumnos se les proporciona rejilla de ítems para que los valoren.

En PIN tenemos 8 actos de evaluación, los cuales se indican a continuación:

- Revisión del Sprint 1 (20%).
- Reunión de Retrospectiva 1 (5%).
- Revisión del Sprint 2 (20%).
- Reunión de Retrospectiva 2 (5%).
- Revisión del Sprint 3 (20%).
- Reunión de Retrospectiva 3 (5%).
- Presentación de la entrega (20%). Co-evaluación entre equipos siguiendo los criterios de evaluación dados por el profesor.
- Auto-evaluación del alumno (5%), basada en una rejilla de ítems dada por el profesor.

En PIN, en las reuniones y en la presentación de la entrega el profesor determina un puntaje para el equipo en base a unos criterios de evaluación establecidos y conocidos por los equipos. Dicho puntaje es distribuido por los equipos para obtener la nota individual de cada miembro. Las notas de

las Reuniones de Retrospectivas las otorga el profesor en base a la reflexión de mejora que realice el equipo, en la cual deben identificar problemas y proponer posibles mejoras, planteadas en términos de reforzar o añadir prácticas ágiles.

### 4. Lecciones aprendidas

Actualmente, en los nuevos estudios del Grado en Informática, tenemos las mejores condiciones curriculares para la aplicación de nuestra estrategia, disponiendo de dos asignaturas consecutivas (semestres 3B y 4A). Además, por estar ambas dentro del mismo módulo de intensificación, los alumnos de dicho módulo deben matricularse en ambas. Sin embargo, esto no ha sido siempre tan favorable y hemos tenido que ir haciéndonos un espacio y consecuentemente evolucionando el enfoque. A continuación comentaremos algunos aspectos de esta evolución, los obstáculos que hemos ido resolviendo y algunas recomendaciones que podemos ofrecer al respecto.

1. ¿Qué hacer si para abordar estos contenidos solo se tiene espacio en una asignatura?. Hemos probado esta situación desfavorable y concluimos que es complicado conseguir en una misma asignatura sintetizar los contenidos de PSW y PIN. Una alternativa menos ambiciosa, que también hemos puesto en práctica, fue centrarnos en la propuesta de PIN, renunciando a proporcionar una visión global de métodos ágiles, centrándose en una configuración de proceso ágil específica (cercana a Scrum). Sin embargo, actualmente nuestra recomendación cuando se tiene solo una asignatura sería llevar a cabo nuestra propuesta para PSW.
2. Existe mucha información asociada a metodologías ágiles y bastantes propuestas concretas de métodos, todas ellas con mucho solape de terminología, conceptos y prácticas. Así pues, en los contenidos hemos ido haciendo cada vez más énfasis en la mezcla de prácticas ágiles y en cómo implantarlas en un contexto específico, en lugar de presentar los métodos como excluyentes o antagonistas. Conseguir esto ha sido una evolución, dirigida principalmente por nuestra experiencia en aplicación real de metodologías ágiles y también con la experimentación docente año tras año.
3. Es fundamental que cada equipo tenga un instructor y un cliente, especialmente este último es fundamental para dar realismo al trabajo práctico realizado por los equipos. Hemos probado con dos profesores realizando cada rol para cada equipo (de forma alterna) y aunque nos pareció el ideal en cuanto a realismo, fue

más difícil de poner en práctica puesto que muchas veces era necesario que estuviesen ambos profesores en la misma clase, lo cual no era siempre posible. Así pues, hemos decidido que en cada grupo el profesor correspondiente desempeñe ambos roles para todos los equipos del grupo. Lo importante es que el profesor como instructor remarque el rol que está desempeñando en cada momento, como también que los participantes soliciten la intervención del profesor explícitamente indicando si quieren soporte del instructor o quieren resolver cuestiones de planificación, requisitos y/o validación con el cliente.

4. Un principio básico ha sido siempre que el profesor se involucre en la orientación de los equipos y la supervisión del proceso. Los equipos podrían llegar a generar un producto al final de la asignatura, prescindiendo de dicho apoyo del profesor durante el proyecto, pero probablemente los equipos se centrarán solo en la programación y en la documentación a posteriori. Si el objetivo es que los alumnos apliquen un proceso de desarrollo explícito es fundamental la intervención constante del profesor. Gracias a que el profesor desempeña el rol de cliente nos aseguramos que existirá un continuo contacto del profesor con los equipos durante el proyecto. Para hacer viable el aumento en la dedicación del profesor en cuanto a interacción con los equipos (como instructor y como cliente), hemos aprovechado la evaluación continua e in-situ para que el trabajo de revisión del profesor se reduzca y se integre con las actividades y trabajo realizado en clase. De forma orientativa el profesor dedica fuera de clases unas dos horas semanales para interactuar con los alumnos, lo cual se realiza a través de la herramienta que utilizamos.
5. La incorporación y aplicación de prácticas ágiles debe realizarse de manera gradual. En PIN no es recomendable que ya en el primer sprint los equipos trabajen con una configuración de proceso muy ambiciosa. Hay que tener en cuenta que al principio tienen más riesgos derivados de su inexperiencia en trabajar en equipo y probablemente en las tecnologías utilizadas. Aprovechando el desarrollo iterativo, en cada sprint se pueden ir introduciendo mejoras en el proceso. Es más, en las retrospectivas posteriores a cada sprint el mismo equipo debe reflexionar y proponer (con apoyo del instructor) las mejoras que se pondrán en práctica en el siguiente sprint.
6. Definitivamente no es recomendable intervenir en la tecnología que cada equipo utilizará para

desarrollar su producto (en PIN). De esta forma, ellos asumirán sus propios riesgos al respecto evitando que recurran al profesor para resolver problemas técnicos. El rol del profesor como instructor es apoyar en cuanto al proceso utilizado, y como cliente debe tomar decisiones respecto del producto, todos los aspectos técnicos deben quedar en manos del equipo. Por contraparte, tratándose de una asignatura de proceso no ponemos mayores inconvenientes en cuanto a que quizás no se utilice la tecnología de desarrollo más apropiada, por ejemplo, que se haga un desarrollo de escritorio cuando lo lógico podría ser un desarrollo web. Normalmente los equipos deciden utilizar tecnologías que han visto en otras asignaturas, aunque también algunos equipos acuerdan utilizar una tecnología dominada por algún miembro que anima a que el resto la aprenda. Por otra parte, si bien advertimos que la arquitectura del producto es importante, destacamos que un proyecto exitoso debe antes que nada cumplir con las expectativas del cliente en cuanto a alcance, plazos y coste. Por lo anterior, desde hace tiempo renunciamos a dedicar tiempo a revisar y evaluar la arquitectura del producto, consideramos que esto excede el tiempo y las competencias de las asignaturas PSW y PIN.

7. La incorporación de una herramienta para la gestión del proyecto contribuye a hacer explícito el proceso y promover una disciplina de trabajo. Si bien somos conscientes que las herramientas son un medio y no un fin, aprender una metodología de desarrollo de software conlleva trabajo en equipo, el cual requiere visibilidad del trabajo, comunicación y colaboración. La situación de los alumnos corresponde con un equipo de desarrollo no colocalizado, que sólo tiene la oportunidad de reunirse cara-a-cara puntualmente unas veces por semana (aprovechando básicamente los horarios de la asignatura). La incorporación y uso de una herramienta para gestión del proyecto y para apoyar la colaboración no presencia entre los participantes (incluido el cliente) se hace imprescindible.
8. En cuanto a espacios docentes, lo más apropiado sería contar con la infraestructura usualmente recomendada para equipos de trabajo en metodología ágiles; mesas amplias, sillas con ruedas, pizarras en el espacio de trabajo, etc. Esto favorecería tanto las actividades realizadas en PSW como el desarrollo del producto en PIN. Hemos pasado por diferentes situaciones, en general disponiendo de un aula para la parte teórica y un laboratorio para la

parte práctica. Si bien no hemos contado con dichas condiciones óptimas, esto no ha sido gran inconveniente. Para llevar a cabo nuestro programa, especialmente en PIN, aprovechamos todas las horas presenciales sin distinguir entre trabajo de aula o seminario, o trabajo de laboratorio. En aula instamos a los alumnos para que traigan sus portátiles, aunque no es indispensable para todo el trabajo del equipo estar con un ordenador.

9. En PIN hemos intentado reproducir los roles planteados por Scrum. Probamos con el profesor como Product Owner y Scrum Master, y el equipo con el rol genérico Desarrollador. Este planteamiento presenta los mismos inconvenientes que en el contexto industrial; el Product Owner debería estar en estrecho contacto con el equipo, debería hacerse cargo de la definición del trabajo y de la planificación de la entrega. Es decir, un buen Product Owner debería ser un Jefe de Proyecto, Analista y Cliente, todo a la vez y en la misma persona. En proyectos reales es muy difícil conseguir dicho Product Owner ideal. Según lo anterior, respecto de la relación entre el cliente y el equipo de desarrollo hemos optado por un planteamiento más realista y sencillo para la asignatura. El profesor desempeña el rol de Cliente y un miembro del equipo tiene el rol de Product Manager, rol que preferimos llamar así para remarcar que tampoco se trata de del Jefe de Proyecto tradicional, pues queremos que el equipo se auto-organice.

## 5. Herramientas de apoyo

En PSW, para ilustrar aisladamente algunas prácticas ágiles utilizamos simplemente papel y lápiz, alguna herramienta sencilla como Trello ([www.trello.com](http://www.trello.com)) o una hoja de cálculo. De hecho cuando comenzamos a trabajar con Extreme Programming y luego con Scrum, las Historias de Usuario las escribíamos en fichas de papel y en hojas de cálculo. Sin embargo, por nuestra experiencia en proyectos reales y en implantación de prácticas ágiles, hemos visto la conveniencia de disponer de una herramienta que permita gestionar dicha información, facilitando el trabajo colaborativo, especialmente para equipos distribuidos.

En 2006 tuvimos la suerte de participar en el desarrollo de una herramienta para la gestión ágil de proyectos, como parte de un proyecto de colaboración con una empresa. Con la experiencia en la implantación y mantenimiento de dicha herramienta decidimos construir desde cero nuestra propia herramienta, llamada Tune-up, la cual está totalmente alineada con nuestro enfoque para implantar

prácticas ágiles. No tenemos espacio suficiente en este artículo para hacer una descripción de Tune-up, sin embargo, tenemos un sitio web con información al respecto: [www.tuneupprocess.com](http://www.tuneupprocess.com). La herramienta se ofrece de forma gratuita para uso académico, incluyendo apoyo para instalación, formación y soporte.

## 6. Conclusiones

Impartir una asignatura de proyecto como PSW o PIN podría simplificarse a: formar equipos de trabajo, asignarles un proyecto y establecer los artefactos que deben entregar al final de la asignatura. El profesor podría limitarse a esperar la reacción de cada equipo, que probablemente no llegará hasta el final del curso cuando los alumnos tengan que hacer la entrega. Esta estrategia extrema puede tener como positivo el hecho que los alumnos probablemente cometerán muchos errores, sin embargo, el profesor no intervendría oportunamente para que los alumnos aprendan a corregirlos.

Nuestra estrategia, con el tiempo se ha ido posicionando en el extremo opuesto al anterior. El profesor, como instructor, interviene constantemente destacando los errores cometidos por los equipos. Gracias a la evaluación continua y al proceso iterativo, los equipos tienen la oportunidad de ir mejorando su forma de trabajo durante el proyecto.

Nuestras colaboraciones en proyectos reales han enriquecido nuestras asignaturas y las ha orientado fuertemente hacia la preparación del alumno para el mercado laboral en lo que respecta a los contenidos abordados. Consecuentemente, el no contar con esta experiencia podría ser un inconveniente para impartir adecuadamente estas asignaturas, pero como contraparte, el alto grado de estructuración y el detalle del material preparado reducen el impacto de dicho inconveniente. Además, no solo dejamos disponible nuestro material, sino que también ofrecemos nuestro apoyo para la implantación de estas asignaturas. Esperamos establecer contactos y colaboraciones docentes con profesores de otros centros de estudio. Asimismo, tal como indicamos antes, ofrecemos nuestra herramienta sin limitaciones para su uso académico, junto con formación y apoyo para su implantación.

Nuestra experiencia también nos ha llevado al convencimiento que, desde el punto de vista pragmático, en la aplicación de metodologías el contexto del equipo es determinante. Si se ignora este hecho existe un alto riesgo que la iniciativa de implantación o mejora de proceso fracase. Dicho contexto normalmente hace inviable el aplicar de forma exclusiva un método ágil específico. Además, cuando ya existe alguna metodología implantada no es fácil prescindir de ella ni tampoco

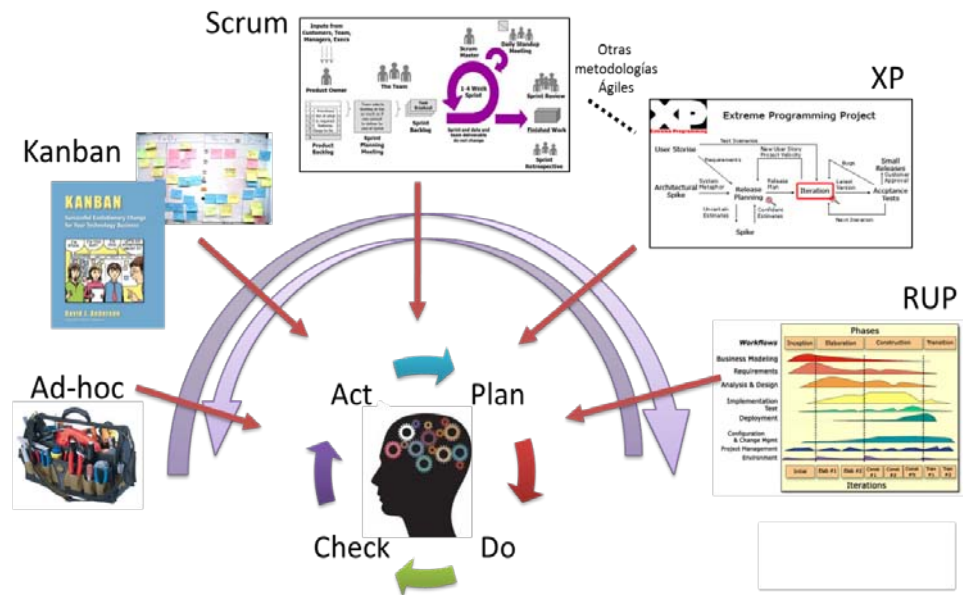


Figura 2: Mezcla y refinamiento de prácticas para un contexto específico

realizar una transición de ella conviviendo con prácticas ágiles. Según todo lo anterior el enfoque de implantación también es un aspecto clave que hemos incluido en nuestras asignaturas. Nuestra propuesta considera un diagnóstico del contexto de trabajo del equipo, la selección de prácticas convenientes para dicho contexto desde un abanico de métodos (Figura 2), posibilitando mezclas de prácticas y su adaptación, todo ello en una hoja de ruta en la cual las prácticas se van incorporando paulatinamente, evaluando en cada paso su efectividad.

Este año, al finalizar PSW hemos realizado una breve encuesta anónima con preguntas enfocadas a conseguir una valoración del alumno respecto de nuestro enfoque docente. La encuesta ha sido respondida por 42 alumnos (sobre un total de 50). A continuación se presentan algunas de las afirmaciones valoradas por los alumnos, indicando en cada caso la suma de porcentajes de respuestas “Totalmente de acuerdo” y “De acuerdo”.

- Los contenidos de la asignatura tienen una clara aplicación en el ámbito laboral (95%).
- Los contenidos de la asignatura han sido eficazmente apoyados con ejemplos y actividades que refuerzan su aprendizaje (88%).
- La asignatura promueve el espíritu crítico respecto de los métodos de trabajo en un equipo (82%).
- La asignatura promueve hábitos de organización del trabajo individual y disciplina de colaboración eficaz con los compañeros de equipo (80%).

- El proyecto de trabajo en equipo permite al alumno recrear con bastante realismo un escenario de trabajo en equipo (81%).
- La asignatura proporciona pautas respecto de cómo debería enfrentarse la implantación de prácticas ágiles en un equipo de trabajo (85%).

Estos resultados confirman que los alumnos han reconocido el enfoque que hemos querido darle a la asignatura.

## Referencias

- [1] David J. Anderson. Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, 2010.
- [2] Kent Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley, 1999.
- [3] Mike Cohn. Succeeding with Agile: Software Development using Scrum. Addison-Wesley, 2010.
- [4] Ivar Jacobson., James Rumbaugh y Grady Booch. The Unified Process Development Process. Addison-Wesley, 1999.
- [5] Ministerio de Administraciones Públicas de España. Metodología Métrica v.3. Disponible en: <http://bit.ly/WZ7ySx>.
- [6] Mary & Tom Poppendiek. Leading Lean Software Development. Addison-Wesley, 2010.
- [7] Ken Schwaber y Jeff Sutherland The Scrum Guide. Disponible en: <http://bit.ly/VyF9ow>.