

Incorporando Extreme Programming como Metodología de Desarrollo en un Laboratorio de Sistemas de Información

Patricio Letelier, José Hilario Canós

Depto. de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

46022 Valencia

e-mail: letelier@dsic.upv.es

jhcanos@dsic.upv.es

Resumen

Este trabajo describe nuestra experiencia en el uso de Extreme Programming como metodología de desarrollo de software en una asignatura de quinto curso en la Titulación de Ingeniero en Informática en la Facultad de Informática de la Universidad Politécnica de Valencia. Se explica el esquema de trabajo utilizado, incluyendo el método docente y el método de evaluación. Por el interés que han despertado en la actualidad las metodologías ágiles pensamos que nuestra experiencia y recomendaciones pueden ser valiosas para asignaturas similares en otras universidades.

1. Introducción

El objetivo de este trabajo es presentar la definición y los resultados de una experiencia docente llevada a cabo durante el primer cuatrimestre del año académico 2002-2003 en una asignatura de quinto año de la Facultad de Informática de la Universidad Politécnica de Valencia (UPV). El trabajo se enmarcó dentro del contexto de programas de ayuda a la mejora de la enseñanza del Proyecto EUROPA [1]. La asignatura objeto de esta experiencia se denomina Laboratorio de Sistemas de Información¹ (LSI). LSI es una asignatura optativa de quinto año en la Facultad de Informática de la UPV para la Titulación de Ingeniero en Informática, y cuenta con una asignación de 6 créditos, todos ellos son créditos de laboratorio.

En cuanto a contenidos, el objetivo de LSI es que el alumno sea capaz de enfrentar un proyecto de desarrollo de software utilizando un enfoque de ingeniería del software desde la perspectiva del triángulo del éxito en desarrollo de software: notación-herramienta-proceso [2]. Esto se materializa mediante la realización de casos de estudio. Algunos de ellos son relativamente pequeños (resueltos en un par de sesiones), trabajando con herramientas CASE (CASE System Architect y Rational Rose) y utilizando la notación UML. La parte central de la asignatura es la realización de un caso de estudio correspondiente al desarrollo de un proyecto siguiendo una metodología. En los años anteriores, se utilizaba exclusivamente el Proceso Unificado de Desarrollo de Software [3] y su versión comercial llamada Rational Unified Process² (RUP). En vista del creciente interés que han generado las metodologías ágiles³ decidimos embarcarnos en una experiencia utilizando una metodología ágil, y después de un breve estudio escogimos Extreme Programming⁴ (XP) [4].

LSI lleva implantada 5 años y ha tenido históricamente una matrícula de alrededor de 40 alumnos. Las clases se imparten en un laboratorio que dispone de 20 puestos de trabajo acondicionados para 40 alumnos. Los alumnos matriculados se distribuyen en 2 grupos, cada uno de los cuales asiste a dos sesiones semanales de 2 horas cada una.

La motivación de este trabajo está basada en el convencimiento de que el esquema de

² www.rational.com

³ www.agilealliance.org

⁴ www.extremeprogramming.org

¹ <http://www.dsic.upv.es/asignaturas/facultad/lsi/>

implantación que hemos seguido y los resultados obtenidos pueden ser útiles para asignaturas similares en otras universidades.

El resto de este artículo está organizado como se describe a continuación. En la sección 2 se explica el interés por utilizar XP en el contexto de LSI. En la sección 3 se describe el plan de actividades y el método docente seguido. En la sección 4 se detalla el método de evaluación utilizado. La sección 5 presenta los resultados obtenidos. Finalmente, en la sección 6 se establecen las conclusiones del trabajo y acciones futuras.

2. XP en LSI

LSI ha ido evolucionando desde un laboratorio en el cual se desarrollaban problemas guiados por el profesor y utilizando herramientas CASE hacia un esquema de trabajo en el cual ha ganado protagonismo la recreación de un proyecto de desarrollo de software mediante trabajo en equipos. En la actualidad se dedica al proyecto alrededor del 70% del tiempo de la asignatura y del correspondiente esfuerzo del alumno.

Por un lado al alumno se le permite tener una experiencia más cercana a la realidad respecto de la ingeniería de software. Pero además se potencian una serie de habilidades personales que por lo general no son cubiertas en otras asignaturas de la titulación, como por ejemplo: trabajo en equipo, desempeño de un rol con determinadas responsabilidades, organización de su tiempo mediante planificación de tareas, negociación respecto de los objetivos a cumplir, participación en reuniones de seguimiento, presentación de resultados, etc. Aunque con RUP habíamos conseguido abordar satisfactoriamente estos dos aspectos, teníamos la impresión de que con una metodología ágil podíamos obtener aún mejores resultados en cuanto a potenciar dichas habilidades personales.

Sin embargo, no queríamos renunciar totalmente al uso de una metodología no ágil (llamadas peyorativamente “metodologías pesadas”) con el fin de ofrecer una visión más global en cuanto a estrategias de desarrollo.

Así pues, decidimos mantener un par de equipos con RUP y establecer 5 equipos con XP. Escogimos XP por ser una de las más populares metodologías ágiles y una de las que se dispone mayor cantidad de material, tanto en la web como

en libros recientemente publicados. Por las restricciones temporales de la asignatura es imposible que el alumno trabaje con metodologías (en dos proyectos) con suficiente detalle durante un cuatrimestre. Para compensar esta posible visión parcial del alumno se establecieron presentaciones introductorias de ambas metodologías con un examen escrito asociado. Además se intentaría realizar comentarios comparativos durante el transcurso del proyecto. Finalmente este último aspecto se vio apoyado porque al comenzar el proyecto detectamos que era necesario que los grupos XP presentaran a la clase sus resultados parciales incluyendo una demo de su producto. En XP no nos pareció adecuado evaluar según los artefactos producidos puesto que éstos eran mínimos y principalmente era el código. Así, se decidió que tanto los equipos RUP como los XP realizarían presentaciones al finalizar cada iteración, además de algunas adicionales de seguimiento.

3. Plan de Actividades y Método Docente

En esta sección nos centraremos en la planificación de actividades relacionadas directamente con el proyecto de desarrollo y particularmente las que afectaban a los equipos XP. En la web de la asignatura puede verse el detalle de todas las actividades realizadas durante el cuatrimestre.

El alumno de LSI tiene 4 horas de clases semanales, repartidas en dos sesiones de 2 horas cada una. El primer día de clases se establecieron los equipos de desarrollo, cada uno de ellos con 6 o 7 integrantes. Los roles asignados fueron: Jefe (“Big Boss”), Encargado de Pruebas (“Tester”, encargado sólo de pruebas funcionales) y a la vez Controlador (“Tracker”) y Programador. El profesor desempeñó el rol de Preparador (“Coach”) para todos los equipos. Normalmente se dispone de unas 24 sesiones efectivas de clases durante el cuatrimestre.

Obviamente, para realizar un proyecto de una envergadura interesante teníamos que renunciar a poder completarlo, pero no es grave puesto que nuestro objetivo se centra más en introducir al alumno en la dinámica del proceso, principalmente el trabajo basado a iteraciones. Así pues se establecieron a priori 3 iteraciones, cada una de 3 semanas (para RUP fueron 1 iteración de

elaboración y 2 de construcción). Más un primer hito de 2 semanas para introducirse a la metodología y para establecer las historias de usuario (en el caso de RUP esto constituyó la iteración de Inicio).

El proyecto abordó la construcción de un sistema de información para una tienda virtual. Sólo se realizó una muy breve presentación del caso de estudio, sin entregar ningún tipo de información asociada. Se planteó que la situación actual del área problema necesitaba de una solución global, incluyendo todos los aspectos de la empresa, tales como: logística, ventas, almacén, transporte, personal, marketing, facturación, etc. Posteriormente, y de acuerdo a negociaciones con el cliente, se acotó el trabajo a los subsistemas de ventas y almacén (en los cuales ya se trabajaba con historias de usuario). Finalmente se consiguió una versión operativa del subsistema de ventas y parcial para el subsistema de almacén.

A continuación se indican las actividades XP realizadas durante el cuatrimestre y la planificación estimada, corregida con algunas pequeñas variaciones que se produjeron en la implantación. No se incluyen las reuniones o sesiones XP realizadas fuera del horario de clases.

Nro. Sesión	Actividad relacionada directamente con el Proyecto de Desarrollo
1	Establecimiento de Equipos Arranque oficial del proyecto
3	Presentación: "Introducción a metodologías ágiles y XP"
4	Control escrito de RUP y XP
7	Presentación de Historias de Usuario y Plan de 1era Iteración
12	Presentación Resultados 1era Iteración y Plan de 2da Iteración
14	Presentación Seguimiento 2da Iteración y Sesión XP
16	Sesión XP
17	Sesión XP
18	Presentación Resultados 2da Iteración y Plan 3era Iteración y Sesión XP
19	Sesión XP
20	Sesión XP
21	Presentación Seguimiento 3ra Iteración y Sesión XP
22	Sesión XP
24	Entrega de Resultados y Evaluación de 3era Iteración

4. Método de Evaluación

En los últimos dos años, en LSI se ha venido aplicando un sistema de evaluación continua multicriterio [5]. Todo el trabajo realizado por los alumnos se evalúa una vez terminada cada actividad realizada. Esta evaluación se realiza durante la sesión de clases, proporcionando los criterios para que el alumno no sólo entienda su evaluación, sino que conozca en el momento oportuno los aspectos en los cuales tiene deficiencias. Para aminorar el esfuerzo en corrección y revisión que esto supondría, se utilizan varias prácticas de evaluación, entre ellas: auto-evaluación, evaluación cruzada entre alumnos y evaluación realizada por los jefes de equipo. Así, el trabajo en la asignatura comprende alrededor de 16 actividades, siendo evaluadas todas ellas con el mismo peso. Entre dichas 16 actividades, más de la mitad son actividades directamente asociadas al proyecto de desarrollo. La evaluación de las actividades aporta el 80% de la nota final. El 20% restante corresponde al examen final (un examen que incluye alrededor de 15 preguntas con respuestas cortas). Las dos últimas sesiones de la asignatura se dedican a un seminario con presentaciones de trabajos voluntarios. Para los alumnos que realizan y presentan un trabajo pueden significar hasta un punto adicional a sumar en su nota final, siendo requisito necesario para quienes aspiren a matrícula de honor (suelen otorgarse al menos una).

El sistema de evaluación continua había sido muy satisfactorio, contando con una acogida favorable de los alumnos (aunque al principio parecen sorprendidos, sobre todo por el hecho que ellos deben participar activamente en la evaluación). El hecho de no tener que acumular para fines del cuatrimestre un montón de documentación y trabajos para revisar, el proporcionar a los alumnos una oportuna valoración de su trabajo, y finalmente el mantener al alumno en un ritmo regular de dedicación a la asignatura, habían merecido el esfuerzo asociado la mayor cantidad de evaluaciones.

La introducción de XP como metodología de desarrollo a primera vista no tenía por qué afectar el sistema de evaluación. Sin embargo, tempranamente nos dimos cuenta que debíamos hacer algunos ajustes. En RUP la evaluación se

centraba en artefactos generados, pero en XP al utilizar mucho menos artefactos y siendo el código el principal de ellos, no nos parecía lo más adecuada seguir con dicha estrategia de evaluación. Decidimos así pasar a un esquema de con presentaciones, en las cuales cada equipo y cada integrante del equipo informaba el estado del proyecto. El jefe informaba de la iteración y su planificación (historias de usuario y tareas abordadas) y de las incidencias ocurridas. El encargado de pruebas y controlador comentaba las pruebas establecidas y las desviaciones en cuanto a las estimaciones. Los programadores hacían una demo del estado del producto. Se hizo una presentación al finalizar cada una de las iteraciones y un par de presentaciones de seguimiento. Al finalizar cada presentación se otorgaban dos calificaciones, una externa asociada a la presentación propiamente tal y a la calidad del resultado parcial obtenido, y otra interna referente a la apreciación del profesor respecto del funcionamiento del grupo y a los objetivos preestablecidos. La primera de estas evaluaciones era una nota igual para cada integrante del equipo. La segunda era una nota que se multiplicaba por el número de integrantes del equipo obteniendo una cifra que se le informaba al jefe para que éste la distribuyera entre los integrantes del equipo. Esto resultó efectivo de cara a ajustar y premiar de acuerdo con las diferencias individuales (juzgadas por el jefe de equipo). Adicionalmente, al final del curso cada miembro del equipo tuvo también la posibilidad de evaluar a sus compañeros y particularmente al jefe. Para los dos equipos utilizando RUP se aplicó el mismo esquema de presentaciones, pero ellas se enfocaron más hacia la revisión y seguimiento de los artefactos utilizados. Con todo esto, se consiguieron 20 calificaciones, eliminándose las 3 peores. Esto último proporciona la flexibilidad necesaria de cara a no forzar en la práctica una asistencia casi obligatoria a todas las sesiones.

Así, con estos ajustes conseguimos mantener nuestra estrategia de evaluación continua y a la vez le dimos un impulso adicional a la participación activa del alumno.

5. Resultados de la Experiencia

En esta sección se resumen los resultados obtenidos haciendo hincapié en los aspectos que puedan servir para experiencias similares y que contribuyan a mejorar nuestro propio esquema de implantación.

5.1 Observaciones Generales

La experiencia de presentaciones y evaluación repartida por cada jefe de equipo resultó interesante y efectiva. Aunque, como se podía suponer, los alumnos en general se asignaron calificaciones buenas. Las evaluaciones desde el jefe hacia los otros miembros jugaron su rol de premio-castigo que no causó problemas destacables en el clima de trabajo en el equipo. La calificación final que recibieron los jefes de parte de los otros integrantes fue alta, lo cual confirmó la impresión de buen trabajo en los equipos XP.

Hay que destacar, que por primera vez (siendo este año el cuarto de aplicación de trabajo en equipos y el tercero trabajando con RUP) se produjeron roces personales entre integrantes de equipos (en los equipos XP). Esto tiene su clara explicación en el hecho que los diferentes ritmos de trabajo y capacidades quedan en evidencia en el esquema de trabajo XP, el cual exige una mayor interacción, participación y protagonismo de los integrantes. Sin embargo, en todos estos casos la situación se apaciguó o al menos se llegó a acuerdos prácticos de cara a los objetivos del equipo. Esto le dio una perspectiva de realismo no esperada al caso de estudio y como tal fue asumida y aprovechada como experiencia.

5.2 Opinión del Alumno

Se realizaron las encuestas docente oficiales pero aún no se dispone de los resultados. Sin embargo, al final del curso el profesor realizó una encuesta anónima en la cual a cada alumno se le pidió que escribiera sus comentarios respecto al método docente, al de evaluación y al enfoque del proyecto de desarrollo. Mayoritariamente las valoraciones fueron muy positivas en estos aspectos. Además, los miembros de equipos XP tenían que opinar acerca de su experiencia con XP usando el siguiente esquema:

1. Para cada práctica de XP debían poner una nota entre 1 y 10 para calificar su grado de aplicación en el proyecto, el grado de dificultad para aprenderla y aplicarla, y la importancia que le atribuían para mejorar la calidad del producto obtenido.
2. Hacer una valoración global de XP.

Práctica XP	Grado de aplicación en el proyecto	Dificultad para aprenderla y aplicarla	Interés que tiene para mejorar el producto final
El juego de planificación	9	4	9
Entregas pequeñas	9	3	9
Metáfora	5	5	6
Diseño simple	8	5	8
Pruebas	7	8	9
Refactoring	6	8	8
Programación en parejas	7	4	8
Propiedad compartida del código	8	6	8
Integración continua	8	5	9
Semana de 40 horas ⁵	8	6	8
Cliente en el sitio de desarrollo	7	5	9
Estándares de programación	5	6	8

La encuesta en su parte XP fue contestada por 25 alumnos. Como era más o menos previsible, las prácticas que se cumplieron en menor grado fueron: metáfora (pensamos que se debe principalmente a lo imprecisa que es su definición), refactoring y estándares de programación. En cuanto al grado de dificultad para aprender y aplicar las prácticas XP, las pruebas y el refactoring destacan como las más difíciles para los alumnos. En general, todas las prácticas XP resultaron de interés para el alumno respecto del resultado obtenido (quizás la única excepción fue la práctica Metáfora, por la misma

razón mencionada antes). Finalmente, la valoración global de XP fue muy positiva.

5.3 Mejoras y Recomendaciones

Aunque la experiencia ha resultado positiva, como es natural hemos detectado algunas mejoras posibles, y que a su vez sirven como recomendaciones para quienes se embarquen en una experiencia similar. Podemos destacar las siguientes:

- a) Realizar al menos una sesión semanal de clases que se corresponda con una sesión XP para subsanar los previsible problemas al principio del proyecto. Al comienzo se tiene un gran inconveniente, los alumnos no conocen la metodología y no se dispone de mucho tiempo como para dedicar sesiones a desarrollar un ejemplo guiado. Las presentaciones introductorias a la metodología mejoraron parcialmente esta situación puesto que mientras no se metieron de lleno en el proyecto los alumnos no se les presentaron dudas. Aunque el primer hito establecido para identificar historias de usuario pretendía darles tiempo para introducirse a XP ese tiempo no fue aprovechado y hasta último momento los equipos no reaccionaron solicitando sesiones XP con el usuario. Posteriormente, durante la primera iteración prácticamente todos los equipos tuvieron dificultades técnicas con los lenguajes elegidos, lo cual unido a su inexperiencia en el trabajo de una iteración hizo que todos los grupos no cumplieran con lo esperado en la primera iteración. Mientras tanto, en clases se desarrollaron actividades independientes del caso de estudio. No fue hasta finales de la segunda iteración en la cual llegamos a dedicar prácticamente todas las sesiones al proyecto. Esta estrategia nos significó enorme esfuerzo en sesiones XP fuera de clases durante la primera mitad del proyecto (alrededor de 4 horas semanales fuera de clases con cada equipo!). Por lo tanto la mejora en este caso será asignar una dedicación uniforme al proyecto contando con una sesión XP desde el principio de la asignatura. Así, cada semana puede dedicarse

⁵ Alrededor de 12 horas semanales en el caso de LSI (incluyendo las 4 horas de clases)

una sesión a actividades independientes y otra al proyecto.

- b) Utilizar el tiempo de dedicación por parejas de programadores como referencia para las estimaciones y negociación de historias de usuario. Para poder ajustar las estimaciones al contexto y limitaciones de dedicación de la asignatura en un principio las estimaciones se hicieron considerando horas reales de trabajo, las cuales posteriormente se multiplicaban por 4 para conseguir un ajuste más realista de acuerdo con la dedicación parcial de los alumnos. Sin embargo, a partir de la segunda iteración se detectó que era más fácil e intuitivo el considerar una supuesta dedicación semanal del alumno a LSI de 12 horas semanales. Así, estas horas se multiplicaban por el número de parejas y por el número de semanas de la iteración para obtener las horas reales disponibles para iteración. Un problema adicional en la primera iteración fue que los equipos no consideraron aspectos tales como: aprendizaje de características no conocidas del entorno de implementación, esfuerzo asociado a realización y corrección de pruebas, y refactoring. Con lo cual en la primera iteración se consiguió en general mucho menos de lo previsto.
- c) El rol de coach (preparador) desempeñado por el profesor resultó muy importante para encauzar el trabajo del equipo y responder a dudas metodológicas, sin interferir mayormente en la autonomía del equipo. Sin embargo, el tener que desempeñar al mismo tiempo el rol de cliente le restó realismo al proyecto. En este sentido sería conveniente que dicho rol de cliente lo desempeñara otra persona. Algunas posibilidades que hemos barajado se comentan a continuación. Se podría conseguir un cliente real, pero tenemos la dificultad que se trata de al menos 7 equipos, lo cual es mucho pedir en términos de dedicación de un usuario real, y además se pasaría más hacia una práctica en empresa haciendo más difícil la formación guiada por el profesor. Otra alternativa más prometedora es intercalar los profesores desempeñando el rol de cliente para algunos grupos y de consultor para otros.
- d) Realización de breves presentaciones de resultados al final de cada iteración y otras adicionales de seguimiento. Como es natural y por falta de hábito, en un principio los alumnos se vieron un tanto agobiados por el hecho que tenían que presentar su trabajo al resto de la clase. Sin embargo, al final del cuatrimestre, después de haber realizado unas 6 presentaciones se pudo constatar la mejora y acostumbramiento en este aspecto. Además, estas presentaciones constituyeron el espacio adecuado para realizar comentarios comparativos respecto de RUP y XP.
- e) También teníamos algunos temores respecto de qué tan apropiadas resultaban las instalaciones donde trabajarían los alumnos, puesto que XP hace mucho énfasis en el espacio de trabajo. Aunque lo ideal era que hubiésemos contado con un laboratorio acondicionado acorde con las pautas XP, no fue tan problemático trabajar en los laboratorios con la infraestructura tradicional. Sin embargo, el mejorar este aspecto no queda descartado.
- f) Aunque en teoría los alumnos tienen la misma formación, es inevitable encontrarse con casos en los cuales no se tiene un conocimiento igual en cuanto a notaciones y herramientas. Esto se debe a paso de Escuela a Facultad, alumnos repetidores o de promociones con otros planes de estudios, cambios en programas de asignaturas previas, etc. En años anteriores se ha probado estableciendo una plataforma de implementación fija (por ejemplo alguna de las siguientes: Oracle Developer y Oracle BD, Visual Basic y SQL Server, o Java y MySQL), la cual debe revisarse en algunas sesiones para conseguir un grado mínimo de conocimiento para todos, pero se corre el riesgo que el alumno se vea en una asignatura de programación o bases de datos, asumiendo que los problemas técnicos son los más importantes y que además deben ser resueltos por el profesor. Otra alternativa que hemos aplicado los dos últimos años es dejar la elección en manos de cada equipo,

existiendo finalmente muchos inconvenientes iniciales por desconocimiento de algunos aspectos del entorno utilizado y también causando problemas en cuanto a las instalaciones en los laboratorios compartidos del departamento. Esta alternativa afectó importantemente a los equipos XP puesto que tuvieron menos tiempo para sobrepasar los problemas de implementación, lo cual impactó en gran medida las estimaciones y objetivos de la primera mitad del proyecto. En este aspecto aún no nos atrevemos decir cual de las dos alternativas es la menos problemática.

6. Conclusiones

En términos generales el resultado ha sido muy satisfactorio. Desde la perspectiva del docente, se ha constatado un incremento en la motivación y participación de los alumnos, los cuales se han sido mucho más protagonistas durante el desarrollo de la asignatura. Esto se refleja en varios aspectos: la casi total asistencia de los alumnos a clases (aunque hay que considerar el hecho que prácticamente en todas las sesiones se realizaba una evaluación o era fundamental para la calificación obtenida), el incremento en asistencia a tutorías, las cuales se transformaron en gran medida en sesiones XP, trasladándose del despacho del profesor a los laboratorios.

Como era previsible, todo este trabajo de mejora docente repercute también y considerablemente en el esfuerzo de dedicación del profesor. Al respecto conviene destacar lo efectivo que resulta incorporar evaluaciones realizadas por los propios alumnos, tanto por su aspecto formador para ellos como para aminorar el trabajo de evaluación por parte del profesor. Por otra parte, el esquema de trabajo y sobre todo el material de apoyo utilizado, en parte ya existían y estaban probados con lo cual no se trataba de un cambio radical, lo que hubiese significado un mayor riesgo.

Finalmente, aunque la experiencia no pretendía realizar una comparativa entre RUP y XP pensamos que vale la pena hacer los siguientes comentarios:

- Con el mismo tiempo de dedicación la funcionalidad implementada fue mayor en los grupos XP. Aunque hay que destacar que en los grupos RUP la primera iteración fue

Elaboración, con lo cual sólo realizaron 2 iteraciones de construcción. Los beneficios de haber dedicado una iteración para establecer la estructura no llegaron a ser apreciados puesto que el proyecto no se terminó. Sin embargo, un par de grupos XP pidieron incorporar como tarea explícita el refactoring en la última iteración, lo cual podría considerarse como un síntoma de problemas en la arquitectura.

- Los equipos RUP se mantuvieron poco activos como grupo y respecto de su interacción con el cliente hasta que llegó el momento de iniciar la primera iteración de construcción, parecían bastante cómodos generando los artefactos establecidos (Visión, Modelo de Casos de Uso, Modelo de Análisis y Diseño, Casos de Prueba, etc.). La primera iteración de construcción RUP provocó un cambio significativo en su actitud y en los modelos que habían realizado. Los equipos XP tuvieron mucho más presión en la primera iteración, pero en las posteriores pensamos que se alcanzó un ritmo y rutina adecuada de trabajo.
- La definición más precisa de los artefactos en RUP y las plantillas proporcionadas ofrecen una guía de la información que debe ser recopilada. Sin embargo, en algunos casos como el documento Visión, se tiene demasiado detalle y repetición de información que produce un efecto negativo. En el caso de XP, los alumnos confeccionaron plantillas para historias de usuario y tareas basándose en los pocos ejemplos que aparecen en los libros pero contenían información que no se utilizó. Respecto a los casos de prueba se les indicó un formato para especificarlas. La precisa descripción de los artefactos en XP podría ser de utilidad.
- En RUP la gestión de requisitos se apoyó utilizando RequisitePro, estableciéndose trazabilidad entre los distintos artefactos. Aunque con sus limitaciones, RequisitePro al parecer resultó útil para organizar los requisitos y artefactos en los grupos RUP. Para los grupos XP no era obligatorio utilizar RequisitePro, de hecho sólo un equipo lo hizo y resultó positivo. En los otros equipos todos los artefactos distintos al código se trabajaron sólo con el

procesador de textos. Así, mientras más artefactos se generaban más difícil se hacía la localización de ellos para revisarlos y para verificar la trazabilidad.

- Por la envergadura del proceso definido por RUP y la configuración a nuestro pequeño caso de estudio, los alumnos se quedan con la sensación razonable de no haberlo conocido completo. En cambio en XP la situación es distinta puesto que han abordado la mayoría de aspectos de la metodología.
- Tal como se reconoce en algunos trabajos publicados, RUP y XP son metodologías alternativas. Sobre todo si consideramos un proyecto de una envergadura pequeña o mediana (relativa y vagamente refiriéndonos a una duración corta en meses y no más de 10 integrantes en el equipo). En este perfil de proyecto encaja nuestro caso de estudio. Sin embargo, como también se ha advertido, la relación contractual y de disponibilidad del cliente es crucial para una exitosa aplicación de XP, hecho que verificamos en el desarrollo del proyecto. Con un poco de esfuerzo conseguimos recrear sesiones de trabajo XP estableciendo citas en los laboratorios en las cuales el cliente pasaba desde media hora hasta más de una hora con el equipo.

Referencias

- [1] Proyecto EUROPA: Una Enseñanza Orientada al Aprendizaje. Vicerrectorado de Coordinación Académica y Alumnado. Específicamente se trata de los programas AME2: “Nuevos Métodos de Enseñanza-Aprendizaje” y AME3: “Mejora de los Sistemas de Evaluación”. Universidad Politécnica de Valencia, 2001. www.upv.es/europa
- [2] Terry Quatrani, Grady Booch. Visual Modeling with Rational Rose 2000 and UML. Addison-Wesley Object Technology Series, 1999.
- [3] Ivar Jacobson, Grady Booch y James Rumbaugh. The Unified Software Development Process. Addison-Wesley, 1999.
- [4] Kent Beck. Una Explicación de la Programación Extrema, Aceptar el Cambio. Addison-Wesley, 2002.
- [5] Patricio Letelier. Una Experiencia en Evaluación Continua Multicriterio Aplicada en un Laboratorio de Desarrollo de Software. Actas de las VIII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2002), pp. 295-302, Cáceres, Julio 2002.