Experiencias en la realización de Estudios Empíricos en cursos de Ingeniería del Software

Félix García, Manuel Serrano, José A. Cruz-Lemus, Marcela Genero, Coral Calero, Mario Piattini
Grupo Alarcos
Escuela Superior de Informática
Universidad de Castilla-La Mancha
13071 Ciudad Real
{Felix.Garcia, Manuel.Serrano, JoseAntonio.Cruz, Marcela.Genero, Coral.Calero, Mario.Piattini}@uclm.es

Resumen

Los estudios empíricos en Ingeniería del Software son fundamentales para la validación de diversos métodos, técnicas, herramientas, etc., y los alumnos juegan un papel fundamental a la hora de llevarlos a cabo. Estos estudios no permiten obtener beneficios centrados exclusivamente en los aspectos de investigación, sino que es muy importante considerar también sus beneficios en la docencia. En este artículo se estudia la aplicación de experimentos controlados en cursos de Ingeniería del Software, destacando los beneficios que estos estudios aportan a los alumnos y a los docentes e investigadores que los llevan a cabo. Además, se presentan los resultados obtenidos en la realización de varios experimentos en cursos de ingeniería del software destacando los importantes beneficios pedagógicos obtenidos.

1. Introducción

Uno de los problemas de la Ingeniería del Software consiste en que a menudo se proponen una gran diversidad de métodos, lenguajes, herramientas, entornos, etc., de los que no se demuestra su utilidad práctica. El mercado competitivo actual, en el que se ha convertido el mundo del software, fuerza a las empresas a buscar la mejora de su calidad. Esta búsqueda supone en muchas ocasiones la adopción de nuevas tecnologías sin constancia de su utilidad práctica dejando de lado otras a pesar de que existen evidencias de su utilidad. Por lo tanto, resulta fundamental que los gestores de las empresas adopten un enfoque de "ingeniería del software basada en la evidencia" a la hora tomar decisiones que pueden resultar beneficiosas para el funcionamiento de la empresa [9].

Debido a esta necesidad, los métodos empíricos han centrado la atención de la comunidad científica en la Ingeniería del Software durante los últimos años. Mediante los métodos empíricos es posible evaluar nuevas aportaciones antes de que sean introducidas en los procesos software de las empresas [15]. Los estudios empíricos más comúnmente utilizados en la Ingeniería del Software son: experimentos controlados, casos de estudio y encuestas, los cuales difieren fundamentalmente en sus objetivos y restricciones. En el entorno académico los estudios empíricos más significativos tanto desde el punto de vista investigador como docente son los experimentos [1].

A la hora de llevar a cabo experimentos, los alumnos juegan un papel muy importante, ya que antes de realizar estos estudios en entornos industriales (lo que requiere un gasto significativo de tiempo, esfuerzo y recursos) en muchas ocasiones los investigadores llevan a cabo estudios piloto con alumnos en entornos académicos [5]. De hecho, hay que considerar que los alumnos constituyen la próxima generación de profesionales [17]. Por ello, los resultados de estos estudios en entornos académicos tienen una gran importancia y permiten obtener conclusiones significativas de cara a su realización posterior en entornos industriales. Bajo ciertas circunstancias, las diferencias entre los alumnos y los profesionales son pequeñas y las tareas requeridas en ciertos experimentos no requieren experiencia industrial, por ello se puede considerar la experimentación con alumnos como viable [14], [2].

Actas del SImposio Nacional de Docencia en la Informática, SINDI2005 (AENUI), pp.155-164
ISBN: 84-9732-443-9

Además, los estudios empíricos no sólo constituyen una aportación científica, sino que también proporcionan importantes beneficios pedagógicos en cursos de Ingeniería del Software, por lo que se establece una importante conexión entre la investigación y la docencia que es fundamental analizar [5].

Hoy en día los alumnos de cursos de Ingeniería del Software empiezan a participar en experimentos, sobre todo en universidades norteamericanas y británicas. Como resultado de ello una gran cantidad de métodos y técnicas han podido ser validadas empíricamente tal y como se refleja en numerosas publicaciones científicas. Sin embargo, muchas de estas publicaciones se centran en presentar los beneficios que los estudios han aportado a la investigación, dejando de lado a los alumnos, lo que incluso hace pensar en ocasiones, que los alumnos han sido "utilizados" de forma egoísta para obtener resultados en la investigación. Es fundamental abordar los beneficios que la experimentación aporta desde el punto de vista pedagógico y aportar estos beneficios a los alumnos cuando se planifican y se llevan a cabo experimentos en entornos académicos.

El principal objetivo de este trabajo es el de estudiar la aplicación de experimentos en cursos de Ingeniería del Software, destacando los beneficios que estos estudios aportan, tanto desde el punto de vista pedagógico como investigador.

En el siguiente apartado se identifican los distintos beneficiarios de los experimentos y las ventajas que estos estudios les aportan, sobre todo al investigador, al docente y a los alumnos. En el apartado 3 se describen las fases que hay que considerar a la hora de llevar a cabo experimentos. Los resultados de diversos experimentos llevados a cabo con alumnos en distintos cursos de Ingeniería del Software, se presentan en el apartado 4. Finalmente se presentan las conclusiones obtenidas y las consideraciones para el futuro.

2. Beneficiarios de la Experimentación en Ingeniería del Software

Cuando se desarrollan experimentos se pueden identificar cuatro beneficiarios principales de los resultados, que tendrán diferentes puntos de vista [5]:

- **Investigador**. Es el encargado de planificar y llevar a cabo el experimento. Su objetivo es demostrar la utilidad práctica de su propuesta u obtener conclusiones preliminares para realizar experimentos en entornos industriales.
- **Docente**. Es el profesor responsable de la asignatura o grupo de alumnos, que constituyen el contexto en el que se realiza el experimento. Su principal objetivo es enseñar los conocimientos y habilidades relacionadas con los estudios empíricos realizados y que van a ser de utilidad a los alumnos en su trabajo como futuros profesionales.
- **Alumnos**. Son los sujetos utilizados en el experimento. Su objetivo es aprender técnicas y habilidades que les puedan servir como futuros profesionales.
- **Empresa**. Las empresas de software son las beneficiarias últimas de los estudios empíricos. Como resultado de dichos estudios, las empresas pueden adoptar nuevos métodos o tecnologías que influirán en la mejora de sus procesos software y en definitiva les permitirán obtener beneficios económicos a medio o largo plazo.

En el contexto académico es de especial relevancia establecer las principales ventajas de los estudios empíricos desde el punto de vista del docente e investigador, que en muchas ocasiones son la misma persona, y sobre todo el beneficio que se aporta a los alumnos. En la Tabla 1 se resumen las ventajas más significativas de llevar a cabo estudios empíricos con alumnos de acuerdo al análisis realizado en [5]:

Tabla 1. Beneficios de los estudios empíricos en entornos académicos

Beneficiario	Beneficios
Docente	Nueva forma de formar a los alumnos respecto de la enseñanza tradicional
	Fomento de la participación en grupo de los alumnos en determinados estudios empíricos
	Mejora de la comunicación con los alumnos
	Nuevas formas de evaluación de los alumnos en situaciones en las que no tienen el estrés típico de un examen formal
	Introducir la ingeniería del software empírica como parte de la enseñanza en la ingeniería del software

Beneficiario	Beneficios
Investigador	Evidencia preliminar para aceptar o rechazar hipótesis
	Demostrar a las empresas software la relevancia de la investigación y la utilidad de llevar a cabo estudios empíricos en sus propias empresas
	Prever los recursos necesarios para realizar experimentos en entornos industriales y preparar el material necesario del experimento para realizarlo en la industria
	Formación de investigadores noveles en el desarrollo de estudios empíricos
Alumno	Formación en materias complementarias a la formación de grado
	Conciencia de nuevos problemas a resolver en el software en general y en las industrias en particular
	Mejor autoevaluación de cuál es su nivel en determinados temas de ingeniería del software que en las clases tradicionales más centradas en los aspectos teóricos
	Percepción de las ventajas de usar métodos empíricos en la ingeniería del software
	Preparación para su futuro profesional en el que en muchas ocasiones serán sometidos a cuestionarios, informes, encuestas, etc.

En la práctica, es muy importante conocer los efectos negativos de llevar a cabo experimentos en entornos académicos, con el fin de que la planificación de los experimentos sea suficientemente cuidadosa para que estos efectos no se produzcan [5]:

- Desde el punto de vista del **investigador** los inconvenientes de realizar estudios empíricos con alumnos son el esfuerzo necesario para prepararlos, ya que nunca deben olvidarse de los aspectos pedagógicos al preparar el estudio. Otro problema son las amenazas a la validez del estudio, tema que se aborda brevemente en el apartado 3.
- Desde el punto de vista del **docente** los problemas que pueden surgir son la necesidad de motivar al docente para llevar a cabo el experimento, ya que le supone un esfuerzo de formación mucho mayor para esa clase que si se tratara de una clase normal y además debe motivar a los propios alumnos creando el ambiente necesario en la clase. El docente debe tener capacidad de atender cualquier duda de los alumnos. Estas necesidades se satisfacen en gran medida si el docente y el investigador son la misma persona.
- Desde el punto de vista de los **alumnos** pueden surgir problemas derivados de pérdida de tiempo, si se trata de experimentos que requieren una formación extensa que les hace perder varias clases en lugar de aprovecharlas formándoles en temas más interesantes o útiles para su futuro profesional. Otro problema potencial es que el resultado del experimento demuestre que la técnica o método que acaban de aprender no es efectivo, aunque en este caso también se puede encontrar la parte pedagógica, consistente en demostrarles que una tecnología o técnica no se puede aceptar sin evaluarla en la práctica por muy nueva o prometedora que sea. Además, en este último caso se pueden intentar hallar las causas del resultado.

En definitiva, es fundamental considerar las ventajas e inconvenientes comentados en la planificación de estudios empíricos, con el fin de que las ventajas sean percibidas por los alumnos y se puedan evitar en la medida de lo posible los problemas que de ello se puedan derivar, como la desmotivación o el descontento. Los autores de este artículo han llevado a cabo como docentes/investigadores experimentos con alumnos en diferentes cursos de la Escuela Superior de Informática de Ciudad Real. Los resultados de dichos experimentos centrados en los aspectos más pedagógicos se describen en el apartado 4. A continuación se describen de forma resumida las principales características y aspectos a considerar a la hora de llevar a cabo experimentos controlados con alumnos.

3. Proceso Experimental

A la hora de realizar experimentos controlados, hay que considerar una serie de factores esenciales para conseguir una buena planificación y un desarrollo satisfactorio, con el fin de obtener resultados que sean creíbles y útiles [16], [23],[18], [3], [17].

Las ventajas de los experimentos es que pueden determinar las situaciones en las que

ciertas afirmaciones son verdaderas y pueden proporcionar el contexto en el que ciertos estándares, métodos y herramientas son recomendables. Sólo si el experimento se realiza adecuadamente, es posible obtener conclusiones sobre las hipótesis planteadas.

Los experimentos requieren ser planificados cuidadosamente si queremos que nos proporcionen resultados útiles y significativos. Por ello es necesario seguir un proceso experimental como el que se propone en [23] que consta de las siguientes etapas: **Definición**, en el que se define el experimento en términos del problema y los objetivos; **Planificación**, donde se determina el diseño del experimento y la instrumentación del mismo; **Operación**, en la que se lleva a cabo el experimento y se recogen los datos empíricos; **Análisis e interpretación**, donde se analizan e interpretan los datos recogidos utilizando técnicas estadísticas; **Evaluación de la validez**, en la que se evalúan los aspectos que pueden amenazar a la validez del experimento (validez de constructo, validez interna, validez externa, validez de la conclusión); y **Presentación y difusión**, en la que se elabora un informe sobre los resultados para facilitar que otros investigadores puedan replicar el experimento.

4. Experiencias de Experimentos con Alumnos en Cursos de Ingeniería del Software

4.1. Trabajos Previos

En este apartado se presentan de forma resumida los resultados de tres experimentos controlados llevados a cabo en asignaturas de Ingeniería del Software y que constituyen los trabajos previos a los presentados en el presente artículo [12]. Los experimentos tuvieron lugar en horario normal de clase, su realización fue voluntaria y se motivó especialmente a los estudiantes para su realización, destacando los beneficios que los experimentos les proporcionan como futuros profesionales. Para conseguir evitar las posibles amenazas a la validez de los experimentos,

- Los sujetos tenían una experiencia y unos conocimientos parecidos.
- Los dominios de los diagramas y modelos del material experimental eran suficientemente sencillos y comunes para facilitar su entendimiento.
- Los esquemas fueron entregados a cada sujeto en un orden diferente para evitar efectos de aprendizaje.
- Los sujetos realizaban por primera vez el experimento (efectos de la persistencia atenuados).
- Los sujetos estaban motivados, dado que los ejercicios formaban parte de los conocimientos que debían adquirir en su formación.
- No se permitió que los sujetos hablaran entre ellos ni que pudieran copiar los resultados unos de otros.
- Todas las dudas fueron resueltas por la persona que supervisaba el experimento.
- Los sujetos no tenían conocimiento, a priori, de los aspectos que se pretendían estudiar ni cuales eran las hipótesis que se habían planteado.

En la tabla 2 se resumen los experimentos llevados a cabo previamente y los resultados obtenidos.

La experiencia obtenida en la realización de experimentos previos ha sido utilizada para la planificación y ejecución de nuevos experimentos con alumnos de ingeniería del software en los que se ha hecho hincapié en los aspectos pedagógicos, sólo llevando a cabo aquellos experimentos acordes con los contenidos de la asignatura y formando a los alumnos en la importancia de los métodos empíricos. Estos experimentos se describen brevemente en el apartado 4.2.

4.2. Nuevas Experiencias

A la hora de llevar a cabo los experimentos y con el fin de evitar en lo posible las amenazas a su validez se han adoptado las mismas medidas descritas en el apartado 4.1.

Tabla 2. Experiencias Previas en la Realización de Experimentos en cursos de Ingeniería del Software

	Métodos de Diseño en Almacenes de Datos [20]	**Modelos de Procesos Software** [10, 11]	**D. Clases UML** [13]
Objetivo	Determinar si los **diseños de almacenes de datos** basados en diagramas de **estrella** son más comprensibles que aquellos realizados basándose en el **diseño tradicional** de **bases de datos**	Establecer la relación entre la **complejidad estructural** de los **modelos de procesos** y su **mantenibilidad** (entendibilidad y modificabilidad)	Averiguar si existe correlación entre la complejidad estructural y el tamaño de los diagramas de clases UML y la entendibilidad y mantenibilidad de los mismos
Participantes	**11 alumnos** de la Escuela Superior de Informática (ESI) de Ciudad Real (UCLM)	**45 alumnos**, 3º curso de Ingeniería Técnica en Informática de Gestión (ESI) y **41 alumnos** 3º curso de Ingeniería Técnica en Informática de Sistemas (ESI)	**24 alumnos** 3º curso de Ingeniería en Informática (ESI)
Material	**3 Diagramas** diseñados según el **diseño tradicional** de bases de datos **3 Diagramas en estrella** (equivalentes semánticamente a los anteriores)	**10 Modelos de Procesos** software representados con el lenguaje de modelado **SPEM** (*Software Process Engineering Metamodel*) [22].	9 Diagramas de clases UML
Modo de Operación	**Realizar consultas** sobre los diagramas del material utilizando **SQL**	En 5 modelos: **Responder 5 cuestiones** relacionadas con los modelos En los 5 modelos restantes: **Realizar 4 modificaciones** en base a nuevos requisitos	**Contestar** el cuestionario adjunto (cinco preguntas) a cada diagrama **Modificar** cada diagrama de clases (cuatro nuevos requisitos)
Datos Analizados	**Tiempos** empleados en realizar las **consultas SQL**	**Tiempos de respuesta** (entendibilidad) y de **modificación** de los modelos (modificabilidad)	**Tiempos de respuesta** (entendibilidad) y de **modificación** de los diagramas (mantenibilidad)
Conclusiones	No hay diferencia en la comprensión de los esquemas debido al método de diseño utilizado y que por tanto es equivalente diseñar un almacén de datos utilizando una metodología tradicional o los diagramas en estrella.	Algunas métricas definidas para evaluar la complejidad estructural de los modelos eran –en cierta manera– válidas, y pueden ser utilizadas como indicadores del tiempo de entendimiento y modificación de los modelos de procesos software	La mayoría de las métricas definidas (complejidad estructural de los diagramas UML) eran –en cierta manera– válidas, y pueden ser utilizadas como indicadores del tiempo de entendimiento y mantenimiento de los diagramas de clases UML
Beneficios Pedagógicos	Los alumnos adquirieron experiencia sobre la realización de experimentos para demostrar empíricamente la utilidad de una técnica	Se impartió un tema especial de Proceso Software en la asignatura de Ingeniería del Software (3º). Ello permitió a los alumnos tener una visión más amplia de la asignatura y aprendieron a modelar procesos	Los alumnos obtuvieron cierta experiencia en el entendimiento y modificación de diagramas de clases UML tareas que podrían tener que realizar tanto en los exámenes como en el desempeño de la profesión
	Los alumnos aprendieron a dudar de los prejuicios y a analizar las causas de que los resultados obtenidos no fueran los esperados ya que los diagramas en estrella son ampliamente utilizados y aceptados para diseño de almacenes de datos	Se dedicó una clase a mostrar los resultados dónde percibieron la importancia de realizar estudios empíricos en Ingeniería del Software	Se les explicó a los alumnos el objetivo de investigación, las métricas definidas, las hipótesis planteadas y los resultados obtenidos. Ello complementó su formación con el aprendizaje de estudios empíricos y medición del software

a) Experimento sobre "Pair-Designing"

Definición

Uno de los aspectos relevantes a estudio en el ámbito de la Ingeniería del software es la transferencia de conocimiento cuando se trabaja en pares. Dentro de las técnicas que favorecen esta transferencia de conocimiento se encuentra la denominada "Pair programming" que es una práctica utilizada en metodologías ágiles consistente en que dos programadores trabajan "hombro con hombro" (*side by side*) en el desarrollo de una misma pieza de código. Un programador, que asume el rol de conductor ("driver") escribe activamente el código mientras que el otro que asume un rol de meramente observador ("observer") identificando defectos y aspectos tácticos y estratégicos. Los roles se intercambian periódicamente. Un beneficio de esta práctica es que se refuerza el incremento de conocimiento de los participantes, en concreto el conocimiento tácito. Esta práctica podría ser aplicada al diseño con los mismos beneficios denominándose en este caso "Pair Designing" en el que el "driver" edita activamente el documento de diseño mientras que el "observer" realiza una revisión continua. Con todo ello se planificó un experimento cuyo objetivo fue demostrar la relación existente entre la aplicación de la práctica "pair designing" y la construcción de conocimiento [4].

Planificación

El experimento se realizó con 42 alumnos de ingeniería técnica en informática de gestión, 39 de ingeniería técnica en informática de sistemas de tercer curso y 12 alumnos de 5º curso de la Escuela Superior de Informática de Ciudad Real. Con el fin de evaluar la construcción de conocimiento durante la aplicación de la técnica "pair designing" los sujetos tenían que mejorar el diseño de un sistema existente en UML. Todos los sujetos tenían conocimientos de modelado del producto con UML pero no sobre metodologías ágiles y en concreto sobre la técnica "pair programming". Por ello se preparó una clase especial en la que se impartieron los conceptos relacionados con las metodologías ágiles y técnicas relacionadas haciendo hincapié en el aprendizaje del "pair programming". Se realizó una pequeña prueba en la que los alumnos resolvieron un ejercicio de especificación de casos de uso aplicando "pair designing". Tras el ejercicio se dedicaron unos veinte minutos a recabar y debatir las opiniones de los alumnos sobre las ventajas e inconvenientes de la técnica. Los alumnos participaron muy activamente en la iniciativa.

El diseño del sistema que los sujetos tenían que mejorar consistía en: una especificación textual con los requisitos, dos diagramas de casos de uso y dos diagramas de clases (un total de 15 clases). Las tareas de mantenimiento que los sujetos debían realizar fueron básicamente dos:

- **Reducir la complejidad**, eliminando entidades (casos de uso, actores, clases, atributos, métodos) o relaciones no significativas para el entendimiento y cumplimiento de requisitos del diseño existente;
- **Mejorar la legibilidad**, modificando las entidades existentes o añadiendo nuevas.

Los sujetos fueron distribuidos equitativamente o de forma individual (32 sujetos) o por parejas (64 sujetos). Para evaluar la variable dependiente (conocimiento del sistema) se prepararon dos cuestionarios QA y QB. Ambos fueron distribuidos a la entrada (antes de iniciarse el ejercicio) y a la salida (tras la mejora del diseño). Cada sujeto recibía un cuestionario a la entrada QA o QB y a la salida recibía el otro cuestionario. Para evaluar los cuestionarios se contaban las respuestas correctas. El proceso experimental a seguir fue:

- Cada sujeto examinaba la documentación de forma individual durante 20 minutos.
- Cada sujeto respondía un cuestionario de entrada durante 10 minutos. El objetivo de este cuestionario era establecer la línea base, es decir, el nivel de conocimiento del sistema antes de trabajar sobre él.
- Los sujetos distribuidos en pares (aplicando "pair designing") y los sujetos distribuidos de forma individual realizaban las tareas de mantenimiento requeridas durante 80 minutos.
- Cada sujeto respondía un cuestionario de salida (10 minutos) para evaluar la construcción de conocimiento tanto para los que trabajan de forma individual como para las parejas.

Operación

Previamente a la realización del experimento se impartió una breve charla en la que se explicó el proceso experimental.

Análisis e Interpretación

Para analizar si el crecimiento de conocimiento era significativo cuando los sujetos trabajaban aplicando "pair designing" respecto a los que trabajaban de forma individual se aplicó el test estadístico de "Mann Whitney". Como resultado se obtuvo que el crecimiento de conocimiento se refuerza de forma significativa al trabajar en pares aplicando "pair designing" respecto a los que trabajan de forma individual. Además este experimento permitió realizar una comparativa con los resultados de un experimento anterior obteniéndose una nueva conclusión de que el crecimiento de conocimiento en pares es más significativo respecto al individual cuando los sujetos emparejados tienen parecido nivel educativo ("background").

Conclusiones

Desde el punto de vista del investigador se encontraron evidencias de que formar pares con individuos del mismo "background" educativo enfatiza los beneficios de la técnica "pair designing" respecto al crecimiento de conocimiento. Comparando los resultados con experimentos previos también se llegó a la conclusión de que agrupar a una persona con "background" científico con otro de "background" no científico no favorece significativamente el crecimiento de conocimiento e incluso podía empeorar los conocimientos del primero.

Desde el punto de vista pedagógico, los resultados del experimento pueden considerarse interesantes para el sistema educativo en las universidades. Tal y como se trata de motivar en nuevas regulaciones como Bolonia, la educación está experimentando un giro del estilo de enseñanza tradicional a un estilo de aprendizaje en el que los alumnos aprenden por sí mismos con la guía de los profesores. En la planificación de este tipo de educación se podrían aplicar prácticas del estilo de "pair designing" para incrementar en la mayor medida de lo posible el crecimiento de conocimiento.

Desde el punto de vista de los alumnos el experimento fue una experiencia muy enriquecedora para ellos:

- Aprendieron un tema adicional a su formación reglada como son las metodologías ágiles y sus técnicas asociadas. Ello les beneficia como futuros profesionales ya que este tipo de enfoques está adquiriendo un auge creciente en las empresas software.
- Tras la realización del experimento se les explicó brevemente los objetivos perseguidos y los resultados, con lo que contrastaron sus opiniones previas sobre la validez de la práctica de "pair designing" con su propia experiencia en el experimento y con los resultados empíricos. Ello les aporta una mayor capacidad crítica a la hora de adoptar nuevos métodos o técnicas sin antes demostrar su validez.

b) Experimento sobre estados compuestos en diagramas de estados UML

Definición

Los diagramas de estados de UML se han convertido en una técnica muy importante a la hora de describir el comportamiento dinámico de un sistema software. En trabajos previos [7] ya habíamos definido un conjunto de métricas para evaluar las propiedades estructurales de los diagramas de estados de UML y las habíamos validado como indicadores tempranos a través de una famila de experimentos [8].

Estos experimentos también habían revelado que, aparentemente, los estados compuestos no influían en la entendibilidad de los diagramas. Este hecho nos parecía un tanto sospechoso por lo que decidimos ir un paso más allá y realizar un experimento controlado y una réplica del mismo centrándonos en el efecto que los estados compuestos tienen sobre la entendibilidad de los diagramas de estados de UML.

Planificación

El experimento fue realizado por 55 estudiantes de Ingeniería Informática de la Universidad de Murcia y la réplica por 178 estudiantes de las

distintas titulaciones (Ingeniería Informática e ingenierías técnicas) de la Universidad de Alicante.

Cabe destacar que la experiencia previa que atesoraban los primeros era notablemente superior a la de los segundos, ya que en las asignaturas relacionadas con la Ingeniería del Software que habían cursado, mientras los sujetos del experimento habían estudiado los diagramas de estados de UML a fondo (incluyendo los estados compuestos), los sujetos de la réplica sólo habían trabajado someramente con los diagramas de estados de UML y no habían estudiado aún los estados compuestos.

Cada sujeto recibía dos diagramas, uno modelado usando estados compuestos y otro sin usarlos, pertenecientes además a dos dominios distintos. Para cada diagrama tenía que responder un conjunto de seis cuestiones para comprobar si había entendido el modelo, anotando el instante en que comenzaban y terminaban de responder el cuestionario.

Como variable independiente tomamos la presencia o no de estados compuestos al modelar los diagramas, mientras que nuestra variable independiente era la eficiencia de los diagramas, medida como la relación entre el número de respuestas correctas y el tiempo invertido en responderlas.

Operación

Antes de que los sujetos realizaran las tareas que se les solicitaban, se les impartió una breve charla en la que se les explicaban las principales características de los diagramas de estados de UML, además de resolver algunos ejemplos del estilo a los que se iban a encontrar durante la realización del experimento.

Análisis e Interpretación

Se utilizó un diseño factorial con interacción confundida para realizar un análisis de la varianza. Este análisis, junto al de los estadísticos descriptivos de la variable dependiente produjeron los siguientes resultados:

- En el experimento, obtuvimos que la utilización de estados compuestos mejoraba notablemente la entendibilidad de los diagramas, ya que los sujetos obtenían mucho mejores resultados.
- En la réplica, las diferencias entre aquellos diagramas que se modelaban usando estados compuestos y los que no, eran muy pequeñas en cuanto a la eficiencia que los sujetos mostraban al entenderlos. De hecho, los resultados para los diagramas modelados sin estados compuestos eran ligeramente mejores. En cualquier caso, en análisis de la varianza de nuevo indicaba que el uso de estados compuestos en diagramas de estados UML está fuertemente relacionado con la entendibilidad de los mismos.

Conclusiones

Como principal conclusión, confirmamos nuestras sospechas de que los estados compuestos son un importante elemento a tener en cuenta al medir la entendibilidad de los diagramas de estados UML.

Un hallazgo que consideramos muy importante es que cuando alguien se enfrenta a la tarea de comprender un diagrama de este estilo, la presencia de estados compuestos ayuda a que la tarea se haga de un modo más eficiente, siempre y cuando se tenga un conocimiento previo del uso de los mismos.

De ahí que consideremos que es ampliamente recomendable que en aquellas asignaturas de Ingeniería del Software que se dediquen al estudio de UML se detengan en el trabajo con estados compuestos cuando se estén estudiando los diagramas de estados.

5. Conclusiones

En este artículo se ha abordado la realización de experimentos en entornos académicos, destacando las ventajas que estos estudios aportan tanto desde el punto de vista del docente/investigador como del punto de vista del alumno. Además, se han presentado los resultados obtenidos en la realización de 2 experimentos en cursos de ingeniería del software en los que se han aplicado las lecciones aprendidas en experimentos previos y que han permitido obtener diversos beneficios pedagógicos.

Uno de los aspectos en los que se ha centrado la planificación, ejecución y comunicación de resultados de los experimentos ha sido la

formación preliminar del alumno en métodos empíricos como una de los paradigmas en ingeniería del software que convierten esta disciplina en una "ciencia" en el que hay que adoptar un enfoque basado en la evidencia a la hora de proponer o adoptar nuevos métodos y tecnologías. Precisamente, lo que distingue la "ciencia" del "arte" es el modo en que los gestores y trabajadores de la empresa toman decisiones en base a argumentos racionales basados en la evidencia obtenida a partir de la experiencia y la investigación y esta visión no es particular a la ingeniería del software, sino que caracteriza a una buena ciencia en general [19]. Siempre que sea posible en asignaturas relacionadas con ingeniería del software se debe inculcar esta visión de la ingeniería a los alumnos, y la realización de experimentos en entornos académicos es una buena forma de conseguirlo.

El entusiasmo mostrado por la mayoría de los alumnos al realizar el experimento y su interés por conocer los resultados obtenidos nos lleva a pensar que este tipo de experimentos son muy beneficiosos y se deberían realizar siempre que sea posible en cursos de Ingeniería del Software, tal y como ya se está realizando en otros centros de estudio internacionales [6], [15].

No obstante, a la hora de llevar a cabo experimentos en cursos de ingeniería del software siempre hay que considerar los beneficios que se pueden aportar a los alumnos y además hay que tener en cuenta diversas consideraciones éticas [21], para que como resultado del experimento no se produzca desmotivación y descontento por parte de los alumnos.

Referencias

[1] Baresi, L., Morasca, S. and Paolini, P. Estimating the Design Effort of web Applications. Proceedings of the 9th International Software Metrics Symposium (METRICS'03), pp. 62-71, 2003.

[2] Basili V., Shull F. and Lanubile F. Building Knowledge Through Families of Experiments. IEEE Transactions on Software Engineering, Vol. 25 Nº 4, pp. 435-437, 1999.

[3] Briand, L., Arisholm, S., Counsell, F., Houdek, F. and Thévenod-Fosse, P. (2000). Empirical Studies of Object-Oriented Artefacts, Methods, and Processes: State of the Art and Future Directions. Empirical Software Engineering, Vol. 4 Nº 4, pp. 387-404, 2000.

[4] Canfora, G., Cimitile, A., Garcia, F., Piattini, M., Visaggio, C. Confirming the influence on educational background in pair-design knowledge through experiments. In Proceedings of The 20th Annual ACM Symposium on Applied Computing (SAC 2005), Santa Fe (New Mexico).

[5] Carver, J., Jaccheri, L., Morasca. S. y Shull. F. Issues in Using Students in Empirical Studies in Software Engineering Education. Proceedings of the 9th International Software Metrics Symposium (METRICS'03), pp. 239-251, 2003.

[6] Carver, J., Jaccheri, L., Morasca. S. y Shull. F. Using Empirical Studies during Software Courses. Experimental Software Engineering Research Network 2001-2003. LNCS 2765, pp. 81-103, 2003.

[7] Cruz-Lemus, J. A., Genero, M. and Piattini, M.: Metrics for UML Statechart Diagrams. In: Metrics for Software Conceptual Models. Genero, Piattini and Calero (eds.), Imperial College Press, UK, 2005.

[8] Cruz-Lemus, J. A., Maes, A., Genero, M., Poels, G. and Piattini, M.: Analyzing Data Extracted from a Family of Experiments for Evaluating UML Statechart Diagrams Understandability. Research Working Paper, University of Ghent (to appear) (2005)

[9] Dyba, T., Kitchenham, B. and Jorgensen, M. Evidence-Based Software Engineering for Practitioners. IEEE Software, pp. 58-65, 2005.

[10] García, F., Ruiz, F. y Piattini, M. "Medición del Proceso Software", VIII Jornadas de Ingeniería del Software y Bases de Datos, Actas de las Jornadas E. Pimentel, N. Brisaboa, J. Gómez (eds), Alicante (España), pp. 303-314, 2003.

[11] García, F., Ruiz, F. y Piattini, M. An Experimental Replica to Validate a set of Metrics for Software Process Models. European Software Process Improvement Conference, Lecture Notes in Computer Science (LNCS 3281), pp. 146-158, 2004.

[12] García, F., Serrano, M., Cruz-Lemus, J.A, Genero, M., Calero, C. y Piattini, M. La Experimentación en la Docencia de Ingeniería del Software. Proceedings de las X Jornadas

de Enseñanza Universitaria de la Informática (JENUI), pp. 237-245, 2004.

[13] Genero, M. Defining and Validating Metrics for Conceptual Models. Ph.D. Thesis Department of Computer Science, University of Castilla-La Mancha, 2002.

[14] Höst, M., Regnell, B. and Wholin, C. Using Students as Subjects – A comparative Study of Students & Professionals in Lead-Time Impact Assessment". 4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE), Keele University, UK, pp. 201-214, 2000.

[15] Höst. M. Introducing Empirical Software Engineering Methods in Education. Proceedings of the 15th Conference in Software Education and Training (CSEET'02), pp. 170-179, 2002.

[16] Juristo, N. and Moreno, A. Basics of Software Engineering Experimentation. Kluwer Academic Publishers, 2001.

[17] Kitchenham, B, Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K. and Rosenberg, J. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, Vol. 28 Nº 8, pp. 721-734, 2002.

[18] Perry, D., Porter, A. and Votta, L. Empirical Studies of Software Engineering: A Roadmap, Future of Software Engineering, ACM, Ed. Anthony Finkelstein, 2000, pp. 345-355.

[19] Pfleeger, S.L Soup or Art? The Role of Evidential Force in Empirical Software Engineering. IEEE Software, pp. 66-73, 2005.

[20] Serrano, M., Calero, C. and Piattini, M. An Empirical Study with Datawarehouse Design Methods, 1st International Workshop Empirical Studies in Software Engineering, Bunse, C., Jedlitschka, A. (eds), pp. 49-54, Finland, 2002.

[21] Singer, J. and Vinson, N. Ethical Issues in Empirical Studies of Software Engineering. IEEE Transactions on Software Engineering, Vol. 28 Nº 12, pp. 1171-1180, 2002.

[22] Software Process Engineering Metamodel Specification; adopted specification, version 1.0. Object Management Group. November (2002). Available in http://cgi.omg.org/cgi-bin/doc?ptc/02-05-03.

[23] Wohlin C., Runeson P., Höst M., Ohlson M., Regnell B. and Wesslén A. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, 2000.