

Experimentos Controlados y Educación en Ingeniería de Software

Sergio G. Zapata, Laura N. Aballay, Gustavo A. Sevilla

Instituto de Informática

Facultad de Ciencias Exactas F. y N. – Universidad Nacional de San Juan - Argentina

{szapata, laballay, gsevilla}@iinfo.unsj.edu.ar

Resumen Mediante los métodos empíricos es posible evaluar nuevos aportes de la Ingeniería de Software antes de ser incorporados en los procesos de desarrollo de software de las empresas. Resulta favorable que la educación superior aplique métodos empíricos en el proceso de enseñanza y formación de los estudiantes. Este artículo presenta el resultado de un análisis, desde la perspectiva educativa, de cuatro experimentos controlados realizados por profesores de la Universidad Nacional de San Juan en asignaturas de Ingeniería de Software pertenecientes a los últimos años de las carreras de Informática de dicha universidad. La utilización de experimentos controlados es una interesante forma de hacer converger las actividades y metas de la docencia con las propias de la investigación, un objetivo siempre buscado en la academia. Es una forma de reunir en una misma actividad ambos roles del profesor, docente e investigador.

Palabras Claves: Educación en Ingeniería de software, Ingeniería de Software, Ingeniería de Software Experimental.

1 Introducción

Las técnicas de Ingeniería de Software requieren ser estudiadas experimentalmente si se pretende que la Ingeniería de Software no sea solo una disciplina teórica. La Ingeniería de Software es una “gran ciencia”; y la experimentación es un ingrediente necesario de cualquier gran ciencia [1].

Mediante los métodos empíricos es posible evaluar nuevos aportes de la Ingeniería de

Software antes de que sean incorporados en los procesos de software de las empresas [2].

Los estudios empíricos más utilizados en ambientes educativos son los experimentos controlados [3] en donde la participación del alumno toma relevancia tanto por ser sujeto de los mismos como destinatario de una acción pedagógica.

Los resultados de estos experimentos pueden ser de gran utilidad para obtener evidencias científicas importantes respecto de un método, técnica o herramienta, que logren confirmar o refutar hipótesis, aportando conocimiento científico a la industria del software.

La formación de profesionales que incorporen nuevos conocimientos basándose en métodos empíricos fiables, como los experimentos controlados, debería ser un objetivo de la educación superior. Habilidades sociales adicionales pueden ser incentivadas en los estudiantes mediante las prácticas experimentales.

Este artículo presenta el resultado de un análisis, principalmente desde la perspectiva educativa, de cuatro experimentos controlados realizados por profesores de la Universidad Nacional de San Juan (UNSJ) en asignaturas de Ingeniería de Software pertenecientes a los últimos años de las carreras de Informática de tal universidad.

Este análisis pone énfasis en el impacto que estos experimentos controlados tiene en los estudiantes y en los profesores, en su doble rol de docentes e investigadores.

En la siguiente sección se desarrolla una descripción breve de cada uno de los experimentos controlados objeto del análisis que se presenta en este artículo. En la sección

3 se expone el análisis conjunto de los experimentos controlados haciendo especial énfasis en los aspectos educacionales de los mismos. En la sección 4 se presenta una discusión sobre los beneficios y debilidades de estos experimentos, y por último en la sección 5 se detallan las conclusiones y los trabajos a futuro.

2 Descripción de los Experimentos Analizados

A continuación se describen brevemente los cuatro experimentos objeto de análisis en este trabajo. Como características principales se exponen los objetivos de cada experimento, el tipo de diseño experimental usado, sujetos que intervienen, modelo instruccional aplicado, contexto curricular utilizado y cantidad de profesores que guiaron el experimento.

2.1 Experimento #1. Comparación de Técnicas de Estimación de Tamaño de Software

Este experimento [4] compara dos técnicas de estimación de tamaño de software, MK-II [5] e IFPUG-FPA [6]. La comparación principalmente se centró en las características de efectividad, facilidad de aplicación e intensidad de uso de cada una de técnicas. Participaron alumnos avanzados de la carrera de grado de Licenciatura en Sistemas de Información de la UNSJ y también, en menor número, egresados recientes de esta misma carrera; 36 sujetos en total. El experimento no formó parte del programa oficial de la carrera, aunque estaba vinculado con temáticas dictadas en la asignatura Calidad de Software de la misma; los profesores de dicha asignatura, en un total de tres, fueron los que lideraron la ejecución del experimento, el cual fue presentado bajo la forma de taller de actualización profesional.

El diseño experimental elaborado fue ad-hoc, sin seguir un método de diseño existente. Esto se debió principalmente a que este experimento fue el preliminar de una serie que tenían como objetivo principal metas educativas y no científicas. No obstante se

respetaron criterios generales que buscaban dar fiabilidad a los resultados obtenidos.

Los resultados conclusivos se obtuvieron en base a mediciones de variables tomadas durante el proceso experimental y en las encuestas realizadas a los sujetos al finalizar el taller.

Estas variables fueron promedio de puntos de función estimados con la métrica MK II y promedio de puntos de función estimados con la métrica FPA. A efectos de lograr la comparación de ambas métricas y poder determinar si estiman igual, se realizó un test de hipótesis.

El modelo instruccional utilizado consistió en una sesión teórica inicial, donde se expusieron los conceptos de cada una de las técnicas de estimación a aplicar; y dos sesiones prácticas, la primera introductoria al uso de las técnicas y la segunda en donde se desarrolló la práctica experimental propiamente dicha que finalmente sirvió para realizar los estudios comparativos, centro del experimento.

Los sujetos alcanzaron los objetivos de aprendizaje establecidos y se pudieron establecer indicios respecto a la efectividad, facilidad de aplicación e intensidad de uso de las técnicas.

Se pudo apreciar, en base a resultados de las encuestas, que la mayoría de los sujetos consideraron que el procedimiento a aplicar para cada una de las técnicas fue expuesto con claridad, como resultado de un entrenamiento previo adecuado. También se pudo observar que los sujetos lograron captar los conceptos enseñados en la sesión teórica ayudándolos a resolver la práctica planteada. En cuanto a la utilidad de la métrica, muchos de los sujetos consideran que la misma es de gran utilidad para aplicarla.

2.2 Experimento #2. Comparación de Técnicas de Lectura de Requisitos de Software

En este experimento [7] se propuso como objetivo comparar la técnica de lectura Basada en Perspectivas (PBR por sus siglas en inglés) [8] frente a una técnica ad-hoc de lectura de requisitos. Se compararon las características

de eficiencia, eficacia, facilidad de uso e intensidad de uso de ambas alternativas.

El despliegue del experimento (diseño experimental) incluyó tres etapas o sesiones; la primera consistió en una sesión de introducción donde se desarrollaron los conceptos teóricos acerca de lectura de requisitos de software. A continuación se llevaron a cabo dos sesiones prácticas en las cuales los participantes aplicaban alguna de las dos técnicas para leer requisitos de software.

La muestra estuvo compuesta por alrededor de 40 sujetos vinculados a la carrera de Licenciatura en Sistemas de Información de la UNSJ, entre estudiantes avanzados y egresados recientes. Fueron tres los profesores que guiaron el experimento. Al igual que el experimento anterior, éste no formó parte del programa oficial de la carrera aunque tuvo fuerte vinculación temática con algunas asignaturas de la misma, aún así se obtuvo una activa participación de alumnos y egresados quienes se mostraron interesados en aprender y practicar nuevas técnicas de Ingeniería de Software.

El diseño experimental intra-sujeto utilizado, basado en la propuesta de Wohlin [9], tuvo dos tratamientos (técnica de lectura PBR y técnica Ad-Hoc) y un solo factor (el proceso de inspección de software), se trató de un diseño experimental simple intra-sujetos, en el que a cada sujeto se le asignó ambos tratamientos.

Las conclusiones obtenidas se basaron en indicadores objetivos obtenidos durante la ejecución del experimento, tales indicadores fueron cantidad de defectos, tiempo consumido, eficiencia (cantidad de defectos detectados por el sujeto sobre el tiempo consumido) y eficacia (cantidad de defectos detectados por cada sujeto sobre la Cantidad Total de Defectos del SRD) por cada técnica. Los resultados que se obtuvieron específicamente respecto a las hipótesis planteadas y analizadas estadísticamente, fueron los siguientes:

- **Cantidad de defectos detectados:** se pudo comprobar que la técnica de lectura PBR detecta más defectos que la técnica de lectura Ad-hoc.

- **Tiempo insumido:** para esta variable, se pudo revelar que la técnica de lectura Ad-hoc insume menos tiempo en su aplicación.
- **Eficiencia:** los resultados estadísticos evidencian una mejor eficiencia la técnica Ad-hoc. Es decir esta técnica obtuvo menor tiempo promedio para encontrar un defecto.
- **Efectividad:** en este caso, la técnica de lectura PBR pudo ser probada como la que logró una mayor efectividad que la técnica Ad-hoc.

También se obtuvieron indicadores subjetivos basados en encuestas realizadas a los sujetos, específicamente respecto de facilidad de uso, preferencia de uso e la intención de uso.

Como puede observarse en (Figura 1), la facilidad de uso promedio de la técnica de lectura PBR es mayor que la de la técnica Ad-hoc.

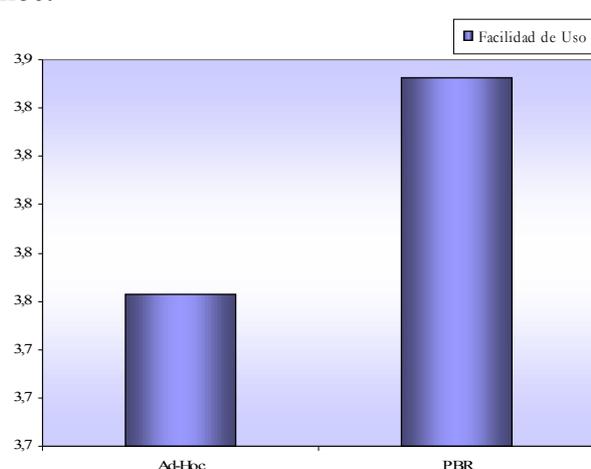


Figura 1. Facilidad de uso de una Técnica de Lectura.

Se estima que la mayor facilidad de uso es justamente debido a que la técnica de lectura PBR guía el proceso, versus la incertidumbre que provoca la TL Ad-hoc, que no tiene procedimiento explícito.

Se observa que 8 de cada 10 inspectores tiene una preferencia hacia la técnica PBR (Figura 2).

La intención de uso muestra a 9 de cada 10 con un pensamiento de uso futuro de la técnica de lectura PBR. (Figura 3).

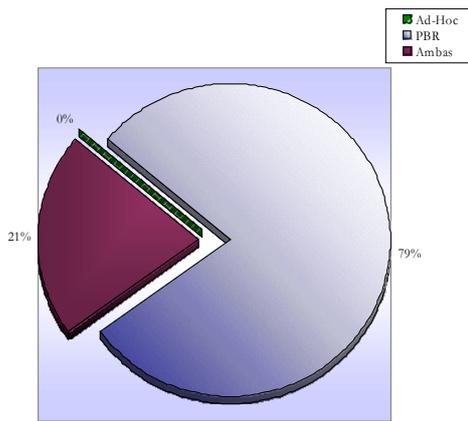


Figura 2. Preferencia de uso de una Técnica de Lectura.

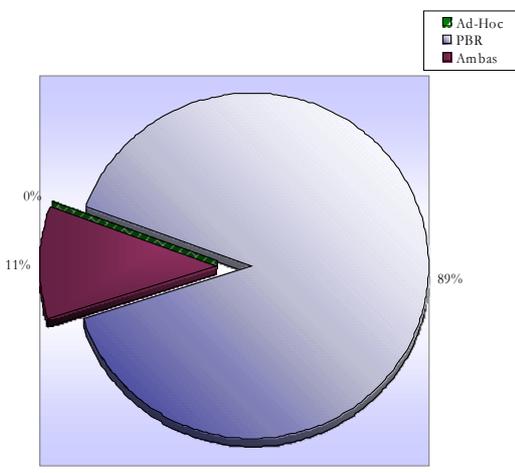


Figura 3. Intención de uso de una Técnica de Lectura.

Respecto a las variables subjetivas, se pudo observar que por parte de los sujetos experimentales hay una inclinación hacia la técnica de lectura PBR.

Al finalizar el experimento los sujetos participantes demostraron haber alcanzado los objetivos pedagógicos, mas allá de los resultados científicos obtenidos en la figura de hallazgos o indicios que deberían ser corroborados mediante repeticiones sucesivas del experimento.

2.3 Experimento #3. Enseñanza de Ingeniería de Software en un Escenario Distribuido y Colaborativo

En el marco de un proyecto de cooperación científica, que vincula a la Universidad del Cauca (Colombia), Universidad de Chile (Chile) y Universidad Nacional de San Juan (UNSJ, Argentina), se ejecutó un experimento

controlado [10] con la participación de alumnos de Informática y Computación de las tres universidades. Los estudiantes participaron de un proceso conjunto de aprendizaje colaborativo aplicando un proceso instruccional, diseñado para tal fin, que utilizó software de comunicación soportado por Internet. La temática objeto del proceso de enseñanza aprendizaje fue Estimación de Tamaño de Software.

Participaron 62 alumnos de los últimos años de las carreras de informática de las universidades participantes; 19 alumnos de Argentina, 22 de Colombia y 21 de Chile. Seis profesores de las distintas universidades participaron guiando el proceso de aprendizaje distribuido.

El modelo instruccional propuesto para este experimento, como se puede ver en la Figura 4, plantea cinco etapas: Preparación, Clase Teórica, Práctica Local, Práctica Distribuida y Evaluación.

En la **preparación** se realizaron actividades de planificación, puesta a punto de la tecnología de comunicación a utilizar y aprestamiento de los grupos de alumnos.

Durante la **clase teórica**, un profesor dictó una exposición introductoria a los estudiantes geográficamente distribuidos utilizando videoconferencia IP; en la **práctica local**, los estudiantes resolvieron un problema práctico trabajando en forma grupal con sus pares locales; luego, en la **práctica distribuida**, los estudiantes resolvieron un nuevo problema práctico conformando grupos de trabajo con pares de otras universidades. Por último, en la **evaluación**, los estudiantes y profesores realizaron una evaluación de la experiencia con el fin de encontrar oportunidades de mejora.

La fase de preparación se realizó en la UNSJ, con la participación de profesores de las tres universidades, los cuales acordaron el contenido temático, el despliegue del mismo, las consignas a implementar en las sesiones prácticas, las tecnologías a usar y finalmente el calendario de actividades.

Un profesor de la Universidad de Chile impartió la clase teórica a distancia.

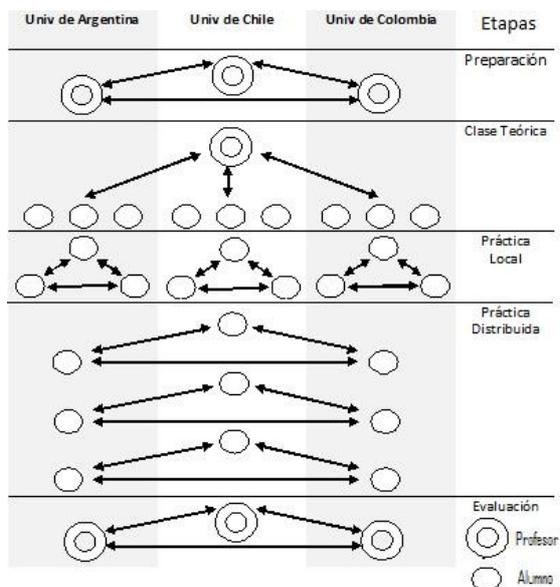


Figura 4. Modelo Instruccional

La tercera etapa, práctica local, se realizó tal cual lo propone el modelo, es decir se conformaron grupos de alumnos de la misma universidad y ellos trabajaron conjuntamente para resolver el problema planteado. Un docente investigador local de cada universidad guió esta práctica.

La práctica distribuida, se llevó a cabo a través de sesiones de trabajo distribuidas. Se definieron 21 grupos de trabajo, conformados aleatoriamente por 3 estudiantes, respetando la condición de uno por cada país involucrado en la experiencia.

En general los alumnos se mostraron muy motivados a interactuar con pares de otras universidades y, más allá de las limitaciones que impone la comunicación mediada por las tecnologías, esa motivación se mantuvo durante todo el proceso.

Los estudiantes tuvieron que “negociar” entre ellos para arribar a acuerdos respecto de las tecnologías a utilizar para la comunicación, las formas de resolver el problema, los horarios de comunicación y la distribución de tareas, entre otros. Estas instancias de negociación los obligaron a poner en práctica habilidades y competencias poco usadas por ellos, pero de importancia para su formación profesional.

Por último se realizó la instancia de evaluación de la experiencia, para ello se necesitó que los alumnos contestaran una encuesta de

satisfacción general del proceso y otra de co-evaluaciones (evaluación entre pares).

Este experimento fue también altamente positivo para los profesores pues se generaron instancias de discusión y acuerdos entre ellos respecto de la enseñanza e investigación en Ingeniería de Software.

En este caso, la experiencia formó parte del programa curricular de la carrera de grado de la UNSJ. La asistencia de los alumnos fue obligatoria y casi el 80% de ellos, según la encuesta realizada, valoró como satisfactoria la experiencia.

La mayoría de los alumnos, el 93%, dijo que aprendieron el tema propuesto, Estimación de Tamaño de Software, de manera que el objetivo pedagógico principal fue alcanzado ampliamente.

La gran mayoría de los alumnos encuestados, casi el 90%, manifestó que sienten haber enriquecido sus habilidades sociales como proponer y sostener ideas ante pares, participar positivamente en grupos de trabajo y adquirir conocimiento de pares. Un aspecto negativo de esta experiencia fue las dificultades de operación (interrupción, lentitud, etc.) de las herramientas de comunicación, las cuales impactaron negativamente en el proceso.

2.4 Experimento #4. Comparación de Técnicas de Testing de Software

El objetivo general de este experimento [11] es obtener una comparación de las técnicas de testing de software, focalizando en los aspectos de eficacia y eficiencia de su aplicación. Las técnicas de testing funcionales comparadas fueron Partición de Equivalencia (PE) [12], [13] y Tablas de Decisión (TD) [13].

Participaron como sujetos del experimento 19 alumnos avanzados de la carrera Licenciatura en Sistemas de Información de la UNSJ, formando el experimento parte del programa oficial de tal carrera. En este caso fueron dos los profesores que guiaron el experimento.

Se aplicó un diseño experimental, siguiendo las consignas de Wohlin [9], que contó con dos factores y dos tratamientos. El primer factor involucra las dos técnicas de testing;

mientras que los dos tratamientos fueron dos programas (códigos ejecutables) que fueron objetos de las pruebas (testing). Cada programa resolvió un problema distinto. Se analizó la aplicación de cada técnica de testing con cada uno de los programas mencionados. Se implementaron una sesión teórica y dos sesiones prácticas las que se desarrollaron en el transcurso de una semana, acumulando entre las tres sesiones aproximadamente 7 horas de trabajo.

Con este experimento se comprobó, analizando los resultados estadísticos que la técnica PE es más eficiente (Figura 5) y eficaz (Figura 6) en el hallazgo de defectos que la técnica TP, independientemente de los dos tratamientos (programas probados).

En función de las encuestas a los sujetos se observó que el 53% tienen intención de uso de la técnica PE, mientras que TD alcanza el 26%. Respecto de la facilidad de aplicación casi el 90% de los sujetos encontró más fácil la técnica PE que TD.

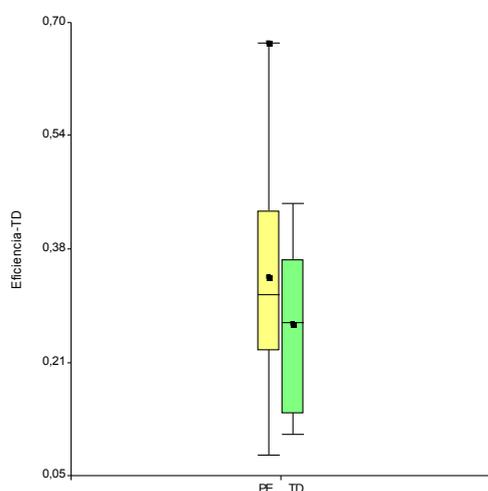


Figura 5. Eficiencia lograda con una y otra técnica

Así este experimento arrojó indicios de interés para la industria del software, específicamente referidos a la efectividad y eficacia de las técnicas de testing. No obstante, es necesario realizar nuevas repeticiones del experimento para afianzar conclusiones.

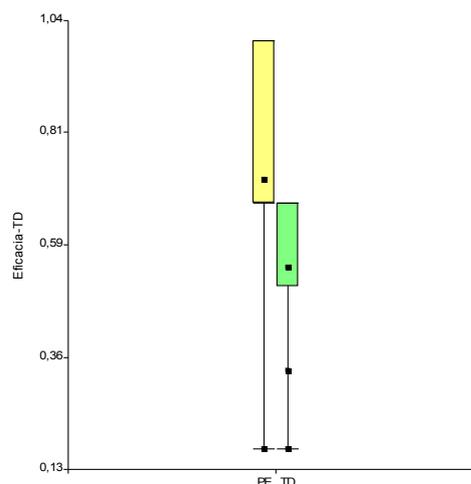


Figura 6. Eficacia lograda con PE y TD

Según las encuestas finales de satisfacción los estudiantes valoraron como “muy buena” la experiencia, destacando que la misma hizo un aporte significativo a su propio crecimiento académico.

3 Análisis de los experimentos

En esta sección se analizan distintos aspectos importantes de los experimentos controlados objeto de análisis en este trabajo. Especialmente se focaliza en la perspectiva de los alumnos, la participación de los profesores y los aspectos de fiabilidad de los experimentos.

En general los alumnos percibieron como beneficioso la realización de este tipo de actividades experimentales. Lo novedoso, al menos para ellos, del modelo instruccional aplicado los predispone de mejor manera para participar; lo nuevo despierta en ellos motivaciones y actitudes positivas hacia el aprendizaje. Muy probablemente lo anterior colaboró, aunque no excluyentemente, para obtener un alto nivel de aprendizaje, como la mayoría de los alumnos manifestaron en las encuestas finales.

El alto componente práctico de las actividades es uno de los aspectos que más satisfizo a los alumnos; esto probablemente se debe a que los estudiantes eran de cursos avanzados y en esas instancias de sus carreras académicas, próximos a la salida laboral, priorizan la adquisición de habilidades prácticas respecto de los conocimientos teóricos.

Los alumnos valoran positivamente la interacción grupal para la resolución de problemas, esta valoración se vio potenciada cuando la interacción se realizó con pares de otras universidades extranjeras. Las diferencias culturales y sociales enriquecieron la formación de los alumnos.

De las encuestas a los alumnos se infiere que los mismos pasaron por instancias de discusión con pares, que los llevaron a poner en práctica y aprender habilidades como la negociación de acuerdos, sostener argumentos, analizar argumentos ajenos, aceptar posturas ajenas, etc. Esto se observa como de gran importancia en la formación profesional de los alumnos más allá de la adquisición de conocimientos técnicos.

En los experimentos en donde se aplicaron estrategias de aprendizaje colaborativas no se pudo medir efectivamente en qué medida estas estrategias aportan al aprendizaje. No obstante, se presume que las mismas son de importante utilidad y se hará un seguimiento más preciso en próximos experimentos para validar estas presunciones. En este sentido hay que remarcar que intentar que estas actividades experimentales incorporen prácticas colaborativas implica un esfuerzo adicional significativo en el diseño del experimento y no siempre es posible lograrlo, por las características propias de las temáticas abordadas.

Los profesores deben realizar un esfuerzo adicional importante en la preparación, ejecución, y análisis de resultados de los experimentos controlados. Se debe tomar especial cuidado en las actividades del diseño experimental, clave para obtener resultados fiables. La selección de los sujetos (alumnos y/o egresados), el tamaño de la muestra, la aleatoriedad en la asignación de los tratamientos, las actividades previas de inducción (clases teóricas), el despliegue de las actividades prácticas experimentales, la definición y medición de variables experimentales y la encuesta final de satisfacción son los aspectos principales a tener en cuenta en el diseño del experimento. En los experimentos analizados se aplicaron

técnicas ad-hoc como también las propuestas de Wohlin [9] para definir el diseño experimental; siendo estas últimas más complejas de implementar pero de mayor confiabilidad respecto de los resultados estadísticos obtenidos.

Los profesores advierten de lo beneficioso de estas propuestas experimentales, aunque coinciden en que las mismas no son aplicables a todos los temas de los programas curriculares, solo a aquellos que tienen mayor independencia con otros temas y además alta factibilidad de ser llevados a la práctica. Las temáticas abarcadas en los experimentos analizados en este trabajo fueron: Estimación de tamaño de software con técnicas de puntos de función, Técnicas de lectura de requisitos basada en perspectivas y Técnicas de testing.

En los experimentos analizados en este trabajo el modelo instruccional aplicado consta de una sesión teórica y dos o tres sesiones prácticas, habiéndose observado que el mismo es muy factible y eficaz en su aplicación. La duración de los experimentos no superó la semana, excepto en el caso del experimento en el que participaron varias universidades que demandó tres semanas debido, principalmente, a las dificultades de coordinación y comunicación de los sujetos. Así, la extensión temporal de estos experimentos es similar a la insumida cuando estos mismos temas se imparten en una modalidad tradicional no experimental.

Se han observado moderadas mejoras en los niveles de aprendizaje, comparando evaluaciones realizadas a los alumnos que trabajaron con procesos instruccionales tradicionales con evaluaciones realizadas bajo esta nueva modalidad experimental. No obstante, uno de los mayores beneficios de los experimentos controlados analizados reside en el aprendizaje de habilidades sociales por parte del alumno, aspecto complejo de medir objetivamente mediante un indicador cuantitativo. Las encuestas finales a los alumnos confirman esta afirmación.

En los experimentos 2, 3 y 4 se alcanzaron indicios significativos sobre técnicas aplicadas en la Ingeniería de Software; especialmente referidas a efectividad, eficacia y facilidad de

uso de las mismas. Estos resultados son fiables estadísticamente, mas allá que por ahora son solo indicios. Este es un aporte importante para la industria del software.

4 Discusión

Claramente se observa que estos experimentos controlados son útiles tanto para los objetivos pedagógicos de la enseñanza de la Ingeniería de Software como para obtener indicios empíricos de valioso aporte científico a la disciplina. Ahora bien, para que estos beneficios se hagan realidad un esfuerzo adicional, al menos inicial, debe ser realizado por parte de los cuerpos docentes. Especialmente en una correcta definición del diseño experimental, cuya importancia es relevante para la obtención de resultados empíricos fiables requiriendo conocimientos específicos sobre Ingeniería de Software Experimental.

La elaboración de paquetes experimentales reutilizables que puedan distribuirse y aplicarse en distintas universidades, mediante repeticiones, no solo disminuiría considerablemente el esfuerzo docente antes mencionado, sino que además potenciaría fuertemente los beneficios científicos de estas actividades experimentales al contar con mayor cantidad de muestras para analizar.

La replicación de paquetes experimentales fortalecerá las conclusiones científicas a que se arriben después de aplicar y analizar los experimentos. En este sentido la conformación de redes de universidades es una acción que puede impulsar fuertemente esta iniciativa.

Para alcanzar la replicación de estos experimentos se deben diseñar paquetes experimentales que sean lo suficientemente generales para soportar los posibles escenarios candidatos de aplicación, siendo esto un desafío futuro relevante. Las universidades latinoamericanas son un potencial dominio de aplicación de estas repeticiones.

Las temáticas de Ingeniería de Software involucradas en los experimentos fueron cuidadosamente seleccionadas, observando que no todas las temáticas son factibles de aplicar. Temáticas de corto despliegue

pedagógico y de alto contenido práctico son las más adecuadas para estas actividades experimentales. Las fuentes temáticas no solo debieran ser los programas curriculares de las asignaturas de las carreras universitarias, sino también la propia industria del software. Esto último favorecería la vinculación entre la universidad y las empresas productoras.

Los alumnos, actores claves del sistema de educación superior, tuvieron una postura favorable hacia los experimentos, según los resultados de las encuestas de satisfacción realizadas en los experimentos analizados en este artículo. Ellos valoran de los experimentos su énfasis en las actividades prácticas, las interacciones con pares de otras universidades y la claridad del modelo de enseñanza aprendizaje. También expresan que es relevante para la exitosa ejecución del experimento un plan de ejecución ordenado y controlado, tecnología de comunicación (chat, email, videoconferencia, etc.) eficiente y alta vinculación temática entre la teoría y la práctica propuesta.

No se observa que los costos monetarios de este modelo instruccional experimental de Ingeniería de Software sean inaccesibles para universidades latinoamericanas. Los recursos necesarios son los estándares con los cuales ya cuentan estas universidades, es decir, acceso a internet, laboratorio de computadores, proyector y tecnología de comunicación IP estándar.

Un aspecto interesante a evaluar es la incorporación de profesionales de la industria del software como sujetos de estos experimentos. No se han advertido resultados distintivos en aquellos experimentos analizados en donde han participado graduados. Tal vez esto se debe a que estos profesionales eran de reciente graduación y la experiencia profesional era similar a la de alumnos avanzados participantes en los experimentos.

Los experimentos controlados llevados a cabo en ambientes industriales son más costosos, principalmente por el costo del recurso humano participante, que aquellos llevados a cabo en ambientes educativos. Una ventaja de

los experimentos controlados analizados es que ellos pueden ser ejecutados repetidas veces en ambientes universitarios hasta alcanzar la madurez de los mismos, momento en el cual pueden ser transferidos a la industria del software para su aplicación en dichos escenarios. Esto incrementa las probabilidades de éxito de los experimentos en ambientes más costosos como los señalados.

5 Conclusiones y trabajo a futuro

La utilización de experimentos controlados es una interesante forma de hacer converger las actividades y metas de la docencia con las propias de la investigación, un objetivo siempre buscado en la academia. Es una forma de reunir en una actividad ambos roles de los profesores, docencia e investigación.

La motivación de los alumnos en estos experimentos controlados es un impulso importante que impacta muy positivamente en el aprendizaje, no solo de aspectos técnicos sino también de habilidades sociales valiosas para su formación integral. También es relevante destacar que los alumnos adquieren, con la aplicación de estos experimentos, los principios de la Ingeniería de Software Empírica que los guía a aplicar mayor rigurosidad científica al momento de incorporar nuevos métodos, técnicas o herramientas en el proceso de Ingeniería de Software.

El modelo instruccional aplicado en los experimentos, basado en sesión teórica y dos sesiones prácticas, ha dado buenos resultados. Sumo cuidado se debe tener en el diseño experimental, en las herramientas de comunicación a utilizar en el caso que hicieran falta y en la incorporación de prácticas colaborativas distribuidas.

Los indicios o hallazgos alcanzados en cada experimentación son un importante aporte a la industria del software, aunque la realización de repeticiones de esas experimentaciones dará mayor validez a las mismas.

La conformación de bibliotecas de experimentos controlados en Ingeniería de Software aplicables a ambientes educativos parece ser el desafío próximo, especialmente

para las universidades latinoamericanas que ya cuentan con las condiciones adecuadas para conformar redes experimentales en esta disciplina.

Agradecimientos. Este trabajo ha sido parcialmente financiado por el proyecto Fortalecimiento Red de Investigación Aplicada en Ingeniería de Software Experimental, en el marco de la Convocatoria de Proyectos de Fortalecimiento Redes Interuniversitarias III del Ministerio de Educación de Argentina. El trabajo ha sido también parcialmente financiado por la Universidad Nacional de San Juan mediante el proyecto Mejora de la Capacidad del Proceso de Software en Empresas Locales (código 21/E 829).

Referencias

1. Basili V.. The Past, Present, and Future of Experimental Software Engineering. *Journal of the Brazilian Computer Society (JBACS)* (2006).
2. 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).
3. 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)
4. Zapata Sergio G., Gomez Claudia P. , Herrera Myriam, Ruiz Susana, Torres Estela, Lund. M. Inés. Diseño Experimental Para Comparar Métodos De Estimación Del Tamaño Funcional Del Software. *X International Conference on Engineering and Technology Education, São Paulo, Brasil* (2008).
5. MK II, Function Point Analysis Counting Practices Manual, United Kingdom Software Metric Association. Version 1.3.1(1998)
6. Abrahão, S. and Pastor, O., Estimating the Applications Functional Size from Object-Oriented Conceptual Models. In *International Function Points Users Group Annual*

Conference (IFPUG'01), Las Vegas, USA. (2001)

7. Lund M. Inés, Herrera Myriam, Aballay Laura, Zapata Sergio. Comparación de Técnicas de Lectura de Documentos de Requisitos de Software: Diseño de un Experimento. *Revista Avances en Sistemas e Informática*, Vol.4 No. 3. ISSN 16577663. pp. 117 - 129. Medellín Colombia. (2007)

8. Basili V., Green S., Laitenberger O., Lanubile F., Shull F., Sørungård S. and Zelkowitz M.V. et al. "The Empirical Investigation of Perspective Based Reading," *Empirical Software Eng.. An Int'l J.*, Vol. 1, No. 2, pp. 133-164. (1996)

9. Wohlin, Runeson, Höst, Ohlsson, Regnell, Wesslén: *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers. (2000)

10.

11. Lund, María Inés, Zapata Sergio, Aballay Laura, Herrera Myriam, Torres Estela, Collazos César, Giraldo Fáber, Ochoa Sergio. *Ingeniería de Software para Ambientes de Aprendizaje Distribuidos*. *Revista Avances en Sistemas e Informática*, ISSN: 1657-7663 Medellín - Colombia. (2009)

12. Soto Ariadna, Sánchez Pablo. *Comparación de Técnicas de Testing de Software* www.idei.unsj-cuim.edu.ar/tesis/tesis_sotosanchez.pdf

13. Myers, G.: *The Art of Software Testing*. New York, Wiley. (1979).

14. Mosley, D. *The Handbook of MIS. Application Software Testing*, ISBN 0-139-07007-9, PrenticeHall .(1993),