

Diseño y desarrollo de prácticas co-evaluables y autoevaluables para la etapa de prueba del desarrollo de software

Inmaculada Medina-Bulo⁺, Pedro Delgado-Pérez⁺, Francisco Palomo-Lozano⁺, Antonia Estero-Botaro⁺, M. Carmen Castro-Cabrera⁺, Juan José Domínguez Jiménez⁺

⁺Departamento de Ingeniería Informática, Escuela Superior de Ingeniería, Avenida de la Universidad de Cádiz nº 10, 11519 – Puerto Real.

inmaculada.medina@uca.es

RESUMEN: En este trabajo se presenta el diseño y desarrollo de prácticas co-evaluables y autoevaluables para la etapa de prueba del desarrollo de software, de manera que ayude a los alumnos a interiorizar la importancia que tiene llevar a cabo pruebas de software de calidad en desarrollos de aplicaciones software reales. Este trabajo también se enfoca en conseguir que los alumnos sean conscientes de los beneficios que esto puede reportar y las ventajas que tiene usar determinadas técnicas y herramientas especializadas para la evaluación y mejora de los conjuntos de prueba, en comparación con un desarrollo puramente manual y sin pautas.

PALABRAS CLAVE: proyecto, innovación, mejora, docente, prueba de software, verificación y validación de software, programación, corrección de prácticas, orientación a objetos

INTRODUCCIÓN

La innovación que se presenta en este trabajo se desarrolla en la asignatura Verificación y Validación del Software, asignatura de tercer curso del Grado de Ingeniería Informática de la Universidad de Cádiz. En esta asignatura se imparten conceptos relacionados con la verificación y validación de software, que abarca todas aquellas actividades conducentes a controlar que se desarrolla un software de calidad. Entre ellas, se encuentra la prueba de software, que tiene como objetivo poder evaluar y mejorar la calidad del software.

En los años en los que lleva impartándose la asignatura, se observa que el alumno encuentra dificultades en entender la importancia de la aplicación de un proceso de prueba de software. Si bien puede llegar a comprender los conceptos y las técnicas empleadas, no queda tan claro que interioricen la necesidad de llevar a cabo dichas prácticas en un desarrollo real o los beneficios que pueden reportar. Por ejemplo, en las asignaturas se les enseña la técnica conocida como prueba de mutaciones, la cual nos permite obtener una medida, conocida como puntuación de mutación, que es indicativa de la capacidad de detección de fallos de la batería de pruebas que se desarrolla. Calcular esta puntuación puede quedar en un mero número para los alumnos si no se le da un significado o aplicación a esa puntuación en un contexto determinado.

Uno de los problemas identificados es que el alumno no es totalmente capaz de discernir la ventaja de usar las técnicas impartidas para la evaluación y mejora de los conjuntos de prueba respecto a un desarrollo puramente manual y sin pautas. Es por ello que pensamos que la comparación directa de prácticas manuales y técnicas como la prueba de mutaciones puede servirles para establecer la necesaria diferencia entre ambas.

Esta es la raíz de la co-evaluación y autoevaluación en las prácticas de la asignatura que presentamos. Esto es, proponemos el empleo de la co-evaluación y autoevaluación

de las prácticas como ayuda para conseguir los objetivos planteados en este trabajo.

Innovaciones similares a esta han sido llevadas a cabo previamente en otras universidades, bien para ayudar al alumno como para la propia evaluación de las prácticas (1) (2). Igualmente, algunos de los autores de este trabajo han presentado también otras propuestas muy relacionadas con esta (3).

OBJETIVOS Y METODOLOGÍA

El objetivo principal de este trabajo es conseguir que los alumnos sean consciente de la necesidad de llevar a cabo pruebas de software de calidad y de la importancia que estas tienen y de los beneficios que pueden reportar a los desarrolladores y usuarios de aplicaciones software.

La idea es que los alumnos actúan primero como desarrolladores y después como evaluadores. En primer lugar, deben desarrollar un conjunto de pruebas de forma manual para un programa dado. En segunda lugar, toman el rol de evaluadores de la práctica realizada por ellos mismos y por otros compañeros. En esta parte se les pide que evalúen su propia práctica (autoevaluación) y la práctica de algún compañero (co-evaluación) según lo bien diseñado y completo que estimen que cada conjunto de pruebas está.

Después se les ofrece los mecanismos adecuados para que puedan aplicar las técnicas impartidas en la asignatura y medir el cumplimiento de ciertos criterios de cobertura (como la puntuación de mutación). Llegados a este punto, pueden comparar esta puntuación con la nota asignada por ellos y observar si concuerdan. Por ejemplo, podría ocurrir que ellos otorguen una alta nota a una práctica y que, sin embargo, la puntuación de mutación sea baja: esa situación les alerta rápidamente de que solo aplicar métodos manuales compromete la calidad del software que se desarrolla.

Para ello, se han llevado a cabo las siguientes actuaciones a través de diversas reuniones:

- Se estableció el número de sesiones en las que se desarrolla la innovación y tramo del curso en el que se aplica (en base al momento del curso en el que se expliquen los conceptos necesarios). En concreto se decidió comenzar a aplicar en dos sesiones y que fueran las últimas del curso.

- Se determinó que las sesiones se llevaran a cabo por parejas.

- Se proporcionó el material necesario para la consecución de la innovación: programa de referencia para la creación del conjunto de pruebas y las herramientas para la medida automática de los criterios de cobertura.

- Se fijó el método de evaluación: porcentajes en que cada criterio, incluyendo el manual, aporta a la nota final de la práctica..

En base a ello, lo siguiente fue el desarrollo del material para poder poner en práctica la innovación en el curso para el que se hace la solicitud. Para la correcta ejecución de esta fase, se creó un repositorio colaborativo de manera que los miembros pudieran realizar esta tarea entre todos.

La siguiente fase fue la de la aplicación de la innovación en las sesiones establecidas y, al finalizar, la cumplimentación por parte del alumnado de la encuesta diseñada.

La última fase, fue la de la evaluación por parte de los profesores de la experiencia y la propuesta de mejora para su aplicación en futuros cursos académicos.

DISCUSIÓN Y CONCLUSIONES

En este trabajo se ha presentado el diseño y desarrollo de prácticas co-evaluables y autoevaluables para la etapa de prueba del desarrollo de software para asignaturas relacionadas con la programación.

Pensamos que esta innovación ha fomentado en los alumnos la necesidad de tener en cuenta los criterios de cobertura que se les imparte a lo largo del curso a la hora de probar aplicaciones software.

Los alumnos valoraron de forma positiva la influencia de este proyecto de innovación en el proceso de aprendizaje de la materia impartida en la asignatura y en la necesidad de realizar pruebas de calidad a las aplicaciones software desarrolladas.

En general, los alumnos recomiendan que se introduzcan conocimientos básicos de prueba de software y, en concreto, de prueba de mutaciones en las asignaturas de programación de primer y segundo curso.

Además de esto, desde el punto de vista del profesor, también nos ha ofrecido datos de cómo se relacionan los criterios de cobertura con la percepción que tenemos de los conjuntos de casos de prueba.

Como trabajo futuro estaría mejorar los programas empleados para la aplicación de la experiencia de manera que, por un lado, favorezcan el comprensión de los conceptos y las técnicas empleadas, y, por otro lado, ayuden a los alumnos a ser consciente de la necesidad de llevar a cabo pruebas de software de calidad y de la importancia que estas tienen, así como, de los beneficios que pueden reportar (tanto a los desarrolladores como a los usuarios de las aplicaciones).

REFERENCIAS

1. Rodríguez del Pino, JC., Díaz Roca, M., Hernández Figueroa, Z., y González Domínguez, JD. Hacia la evaluación continua automática de prácticas de programación. En Actas de las XIII Jornadas de Enseñanza Universitaria de Informática (JENU). 2007, páginas 179-186.
2. Romero, FP., Serrano-Guerrero, J. y Pérez de Inestrosa, H.. Cuestor: Una nueva aproximación integral a la evaluación automática de prácticas de programación. En Actas de las XVI Jornadas de Enseñanza Universitaria de Informática (JENU). 2010, páginas 493-500.
3. Delgado-Pérez, P. y Medina-Bulo I. Automatización de la corrección de prácticas de programación a través del compilador Clang. En Actas de las XXI Jornadas de Enseñanza Universitaria de Informática (JENU). 2015, páginas 311-318.

AGRADECIMIENTOS

Al alumno colaborador Miguel Ángel Sánchez García por su inestimable ayuda en el desarrollo de este proyecto.

Al Vicerrectorado de Recursos Docentes y de la Innovación por la confianza y financiación concedida para realizar este proyecto.