

Un experimento sobre hábitos de pruebas artesanales de software: Resultados y Conclusiones

Pedro J. Lara Bercial, Luis Fernández Sanz

Departamento de Sistemas Informáticos

Universidad Europea de Madrid

pedro.lara@uem.es, luis.fernandez@uem.es

Resumen

En el mundo del desarrollo de software, los retrasos en las etapas previas hacen que sea frecuente que el tiempo dedicado a pruebas sea más reducido incluso que lo escasamente planificado inicialmente. Desde esta perspectiva, resulta de vital importancia conocer cómo piensan y abordan las pruebas los desarrolladores peculiarmente cuando adoptan una filosofía no sistemática de diseño de casos. Así, se aprovechó la oportunidad de otras investigaciones realizadas por los autores en el área de la automatización de pruebas para incrementar el conocimiento sobre otros aspectos prácticos implicados en las pruebas. De esta forma, además de la información necesaria para demostrar de AQUABUS (nuevo algoritmo de generación automática de pruebas a partir de especificaciones UML), se obtuvo información relevante sobre cómo los diferentes profesionales del desarrollo y gestión de proyectos de software llevaron a cabo el diseño de pruebas de un software específicamente pensado para evaluar su proceder y actitud frente a las pruebas planteadas.

1. Introducción

AQUABUS¹ [1] es un algoritmo diseñado por los autores para generar casos de prueba a partir de especificaciones UML; en concreto, a partir de un tipo especial de diagrama de actividad que incluye información adicional para permitir la priorización de los casos generados en función del riesgo derivado de dejarlos sin probar. La

validación de este algoritmo como parte de la tesis doctoral de Pedro J. Lara [2] incluyó también esta serie de objetivos:

- Validar la importancia de una buena especificación
- Validar la compleción de AQUABUS en cuanto a la consecución de una lista suficiente de casos de prueba y la necesidad de priorización de los mismos.
- Validación de las posibilidades de aplicación de AQUABUS en proyectos reales.

Para cada objetivo se llevo a cabo una actividad o experimento diseñado expresamente para el mismo, aunque en algunos casos, los resultados de una determinada actividad sirvieron no sólo para apoyar el objetivo perseguido, sino también para facilitar la consecución de alguno de los otros objetivos.

En esta comunicación se explica brevemente en qué consistieron dichos experimentos y se detallan aquellos resultados que aportan información relevante para entender cómo los profesionales del desarrollo de software suelen llevar a cabo las tareas de prueba. Dicha información incluye resultados acerca de

- Las diferencias entre diseñar pruebas con una buena especificación y una mala.
- Comparación de datos cuantitativos acerca del número de pruebas realizadas manualmente, frente al número de pruebas posibles (estimado con AQUABUS)
- Datos acerca del número de veces que se repiten las mismas pruebas o pruebas equivalentes.
- Comparativas entre las pruebas que se creen a priori más importantes y las que se realizan en realidad.
- Valoraciones acerca de la rentabilidad de un automatismo de pruebas.

¹ Del Inglés: "Algorithm for Quality Assurance Based on UML Specification"

2. Descripción de las actividades de validación de AQUABUS

En la investigación original [2] se perseguía, en resumen, comprobar las siguientes afirmaciones:

1. El diseño de casos de prueba es más completo en cuanto más completa sea la especificación del sistema.
2. El método propuesto (AQUABUS) es capaz de generar al menos todos los casos de prueba relacionados con datos que diseñaría un experto en desarrollo de software de manera artesanal.
3. El estudio del riesgo derivado del hecho de dejar una determinada prueba sin realizar implica un esfuerzo de análisis que ayuda a la correcta priorización de las mismas, reduciendo las probabilidades de estar probando inconscientemente aquello que es menos importante, aunque más evidente.
4. El esfuerzo de integración de herramientas de aseguramiento de calidad y del método propuesto en un proceso de desarrollo no solo es posible desde el punto de vista teórico, sino que además es rentable de cara a las organizaciones desde el punto de vista práctico.

Para conseguirlo se diseñaron las actividades experimentales que se describen a continuación.

2.1. Validar la influencia de la especificación

Para evaluar en qué medida afecta el tipo de especificación de partida al resultado final de la fase de pruebas, se diseñó un primer experimento que consistía en la realización de las actividades que pueden verse en la Figura 1 y que se explican a continuación:

1. Se entregaron las especificaciones de una aplicación sencilla en tres formatos distintos a sendos grupos distintos de programadores (fundamentalmente alumnos de 4º y 5º de Ingeniería Informática y 3º de Ingeniería Técnica de Gestión y Sistemas):
 - Una especificación meramente textual.
 - Una especificación basada en una plantilla de Casos de Uso pero sin diagramas UML adicionales.
 - Una especificación basada en una plantilla de Casos de Uso que incluía diagramas UML de actividad.
2. Cada programador, a partir de la especificación que les fue asignada en función del grupo al que pertenecía, se encargó de diseñar los casos de prueba.
3. Se compararon los casos de prueba generados por cada grupo con el fin de validar la afirmación de que con una mejor especificación de requisitos se consiguen mejores diseños de prueba (afirmación número 1).

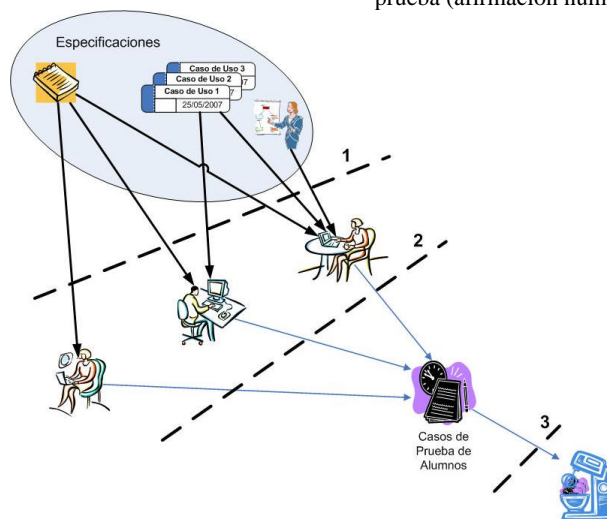


Figura 1. Esquema de validación de la especificación

2.2. Validación de AQUABUS

A la hora de validar el método que se proponía se pensó, fundamentalmente, en dos objetivos que coinciden con la segunda y tercera afirmación de las comentadas anteriormente. Para ello se llevaron a cabo las siguientes actividades previas:

1. Se diseñó e implementó una aplicación PHP de gestión de una base de datos de DVDs con el fin de que pudiera ser ejecutada a través de Internet desde cualquier navegador.
2. Se construyó una librería PHP para registrar en una base de datos, el paso por cada una de las actividades incluidas en los diagramas de actividad utilizados durante el diseño, aplicable a cualquier aplicación PHP y no sólo a la construida para esta validación.

3. Se añadieron, a dicha librería, funciones para registrar los datos introducidos por el usuario durante la ejecución de la aplicación.

4. Se integró la aplicación en un conjunto de páginas web desde las cuales se explicaba al usuario cómo utilizar y probar la aplicación.

Una vez construida la plataforma PHP de validación se llevo a cabo el experimento según se describe a continuación (Figura 2):

1. En la página principal se mostraba al participante una breve introducción del marco de la experiencia junto con la especificación de la aplicación a probar y se le daba paso a iniciar la misma.
2. Una vez que el participante decidía empezar, se le explicaba cómo realizar adecuadamente el experimento.
3. Después se le pedía rellenar un formulario con una serie de información básica que servía para identificar el perfil profesional del participante.

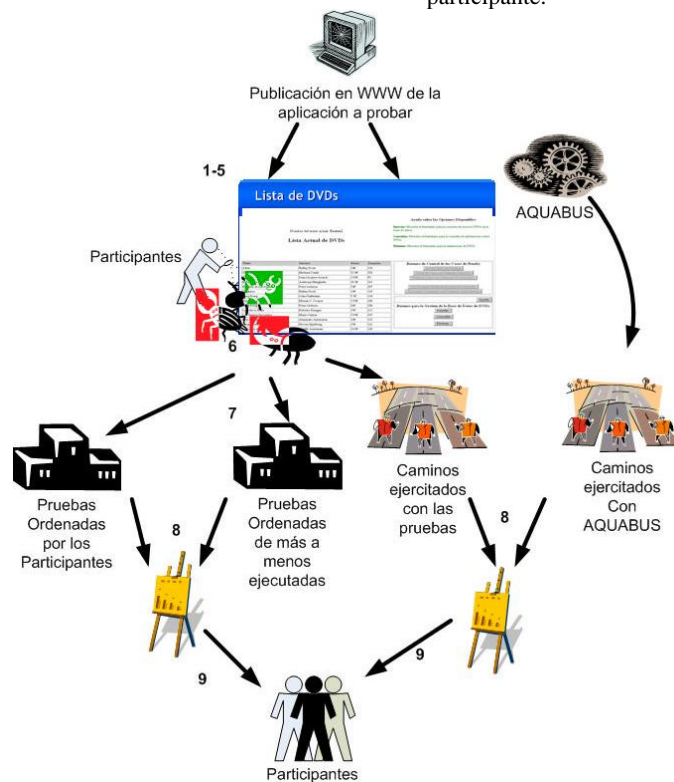


Figura 2. Esquema de validación del método AQUABUS

4. Una vez registrada la información anterior, el participante podía comenzar a realizar las pruebas sobre la aplicación desde una pantalla como la de la Figura 3 en la que tenía la opción de probar los tres tipos de casos de uso (inserción, consulta y borrado) de la aplicación.
5. Según se realizaban las pruebas, y dado que éstas se almacenaban en la base de datos, el participante podía consultarlas en cualquier momento con objeto de saber qué había probado ya y qué no. En un principio, se pensó en no darle esta opción al usuario puesto que era una manera de dar una herramienta de registro de pruebas de la que, en muchos casos, no se dispone cuando se prueba una aplicación.
6. Finalmente, se optó por la opción de dar un mínimo de información acerca de las pruebas que se iban realizando, para descargar de algo de trabajo a los participantes; uno de los principales riesgos del experimento era la posibilidad de que muchos de los participantes no lo acabaran porque les supusiese demasiada dedicación.
7. Cada vez que el participante realizaba una prueba debía indicar si el resultado de la prueba era o no exitoso; es decir, debía indicar si el sistema se había comportado o no tal y como se esperaba que lo hiciera (se habían insertado a propósito varios defectos de funcionamiento en el comportamiento del

programa). Esto da origen a varios tipos de resultados:

- Casos evaluados por el participante como pruebas que detectaron defectos y que, sin embargo, no lo hicieron porque la especificación así lo determina explícita o implícitamente.
 - Casos evaluados como pruebas que no detectan defectos y que, sin embargo, las especificaciones indican lo contrario. Por ejemplo, el hecho de no poder insertar un DVD con precio 0, cuando las especificaciones indican que sí debe permitirse para representar el caso de DVDs que se regalan con otras compras.
 - Casos evaluados correctamente por el participante, tanto en sentido positivo o negativo, en cuanto a la detección de defectos.
8. Cuando el usuario decidía que la aplicación estaba suficientemente probada, finalizaba la primera parte de la prueba y se le pedía que priorizase las pruebas que él mismo había realizado con el objetivo de evaluar si realmente se prueba con más insistencia lo que se considera más importante o más peligroso.
 9. Se mostraban los resultados obtenidos y la comparación con la solución “buena” generada con AQUABUS.

Ayuda sobre las Opciones Disponibles

Insertar: Muestra el formulario para la creación de nuevos DVDs en la base de datos

Consultar: Muestra el formulario para la consulta de información sobre DVDs

Eliminar: Muestra el formulario para la eliminación de DVDs

Nombre del tester actual: Tester2

Lista Actual de DVDs

Título	Director	Precio	Duracion
Alien	Ridley Scott	12€	116
Casablanca	Michael Curtiz	13.9€	102
El Oso	Jean-Jacques Annaud	5.99€	93
El Paciente Inglés	Anthony Minghella	18.9€	165
El Señor de los Anillos	Peter Jackson	1.5€	587
Gladiator	Ridley Scott	19€	110
King Kong	John Guillermin	7.5€	134
King Kong	Merian C. Cooper	5.99€	100
King Kong	Peter Jackson	21€	180
La Vida es Bella	Roberto Benigni	17€	115
Los Santos Inocentes	Mario Camus	5.99€	105
Mar Adentro	Alejandro Amenabar	12€	125
Parque Jurásico	Steven Spielberg	17€	121
Un Lugar en el Mundo	Adolfo Anistaraín	14.9€	120

Botones de Control de los Casos de Prueba

Botones para la Gestión de la Base de Datos de DVDs

Figura 3. Pantalla principal de prueba

2.3. Validación de la aplicabilidad

Para validar la necesidad y la aplicabilidad de un método automático de diseño de pruebas y en particular de AQUABUS, se pidió a los mismos participantes en el experimento anterior que respondieran un cuestionario que incluía dos tipos de preguntas:

- unas relacionadas con el modelo de diseño, ejecución y gestión de pruebas seguido en el entorno de trabajo del experimento
- otro grupo de cuestiones orientado a evaluar el grado de aplicabilidad y rentabilidad de la automatización del diseño de pruebas.

3. Resultados relevantes

Como ya se ha dicho al principio, los resultados que se muestran a continuación son un subconjunto de todos los obtenidos (ver [2]). En concreto, corresponden con aquellos resultados de los que se obtiene información adicional a la meramente necesaria para la validación del método que se proponía como objetivo inicial.

3.1. A mejor especificación mejores pruebas

Uno de los primeros datos, cuyo análisis va más allá de lo relacionado con AQUABUS, son los obtenidos durante el análisis de la influencia de la especificación en las pruebas. En este caso se comprueba claramente cómo una mejor especificación que sirva como punto de partida para la generación de casos de prueba, da como resultado un diseño más completo. La muestra de participación en esta fase de análisis está especificada en la Tabla 1 para los distintos grupos de trabajo creados para la comparación de resultados.

Se observa, en la Figura 4, que **cuando se dispone del diagrama de actividad** como parte de las especificaciones (grupo 3), **se produce un aumento ostensible tanto en el número de casos de prueba diseñados** (en color azul) **como en el número de valores probados para los datos** (en color granate).

Tabla 1. Datos de la muestra de participación

Nº participantes	28
Nº particip. Grupo 1. (Espec. Textual)	8
Nº particip. Grupo 2. (Espec. Casos Uso)	8
Nº particip. Grupo 3. (Espec. Diag. Actividad)	12
Casos de prueba sin repeticiones ² y sin incluir combinaciones de valores ³	14
Total de posibles casos de prueba sin repeticiones y sin incluir combinaciones de valores	15
Total de casos de prueba con repeticiones ⁴ y sin incluir combinaciones de valores	52
Total de casos de prueba con repeticiones, incluidas las combinaciones de valores de datos ⁵	569
Valores utilizados en alguna combinación de prueba sin repeticiones	19
Total de valores posibles en alguna combinación de prueba sin repeticiones	35
Valores utilizados en alguna combinación de prueba con repeticiones	62

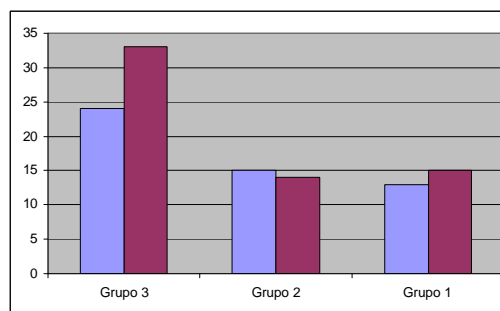


Figura 4. Datos obtenidos.

Por otro lado, al contrario de lo que hubiera podido pensarse, la diferencia entre tener una simple especificación textual (Grupo 1) y una especificación basada en casos de uso algo más elaborada (Grupo 2) no es demasiado importante si ésta no incluye diagramas de actividad.

² Contando una sola vez aquellos casos descubiertos por más de un participante

³ Contando una sola vez aquellos casos que aparezcan varias veces con valores de prueba distintos para los datos

⁴ Contando cada caso tantas veces como participantes lo hayan diseñado

⁵ Contando cada caso tantas veces como combinaciones de valores se hayan diseñado para el mismo.

3.2. Prueba no sistemática: escasa y sin criterio

En este mismo experimento, se puede observar que frente al valor de referencia (total de casos generados con AQUABUS) los alumnos del grupo 3, en media, alcanzan el 71% del total de casos generados con el método, mientras que los de los otros dos grupos se quedan por debajo del 50%. Es decir que **la prueba no sistemática, incluso con una buena especificación tampoco alcanza un nivel adecuado.**

Cuando se pide a alguien que diseñe las pruebas para una aplicación, tal y como se le planteó a los participantes del primer experimento, implícitamente se está obligando al *tester* a que se pare a pensar, analizar y diseñar las pruebas; de otra manera, especialmente en una aplicación simple como ésta, seguramente nunca nadie diseñaría sino que las pruebas serían realizadas directamente “al vuelo”.

Por este motivo se realizó el segundo experimento, en el que simplemente se les pedía probar, sin hablar en ningún momento de diseñar.

Los resultados son aún más significativos. De un total de 71 *testers*, tan **sólo 1 probó más del 75% de los caminos de prueba obtenidos por AQUABUS**, mientras que **el 56% de los participantes se quedaron en menos de la mitad de caminos.**

Otro resultado curioso es que, en media, **la mitad de las pruebas realizadas, probaban caminos ya probados previamente por ese mismo tester.**

Dado que también se les pedía priorizar sus pruebas, indicando cuáles realizarían en el caso de no tener tiempo o recursos para realizarlas todas, se obtuvieron resultados acerca de la coherencia de esta priorización. Comparando la lista de casos de prueba ordenada por prioridad, con la lista de casos de prueba ordenados por el número de veces que se repitió la prueba de dicho caso, se confirmó que a diferencia de lo que se pudiera pensar, **entre los diez caminos más probados, tan solo aparece uno de los diez más prioritarios** (en el puesto octavo). Pero es que **entre los diez caminos menos probados aparecen tres de los 10 caminos valorados como más prioritarios.**

3.3. Automatización de Pruebas: Necesario y, en muchas ocasiones, rentable

Por último, de las respuestas a las preguntas de la encuesta, se obtuvieron los siguientes resultados en relación con la aplicabilidad y rentabilidad de la automatización de pruebas (la muestra es la descrita en la Tabla 2):

Tabla 2. Distribución de la muestra por perfiles

Perfil	%	Media de años de experiencia
Programadores	4%	2.5
Analistas	3%	4.5
Ingenieros de Software	6%	3.3
Jefes de Proyecto	8%	5.5
Testers	13%	3
Profesores de Ingeniería del Sw	3%	10.5
Profesores de Tecn. Información	10%	8.4
Estudiantes de últimos cursos	40%	
Otros	13%	

- Hay un 44% de encuestados que utiliza diagramas UML para diseñar, más un 21% que lo hace dependiendo del proyecto.
- Un 64% no utiliza UML para diseñar pruebas en modo alguno. Siendo un 55% el que no utiliza tampoco ningún otro método definido de diseño de pruebas.
- El 77% de los encuestados considera que a pesar de tener que elaborar diagramas adicionales para cada caso de uso la aplicación de algún tipo de método automático en sus organizaciones seguiría siendo rentable.
- El 70% considera rentable realizar una priorización previa sobre los casos generados.
- El 79% de los participantes consideran útil o muy útil el uso de un mecanismo que automatice el diseño y ejecución de las pruebas, siendo un 9% de los encuestados los que lo consideran imprescindible

4. Conclusiones

Como parte de una serie de experimentos de validación de un nuevo método automático de generación de casos de prueba, se ha podido obtener información valiosa sobre cómo los desarrolladores de software plantean las pruebas. Se ha podido constatar la influencia de una buena especificación en el diseño de pruebas a la vez que las limitaciones de eficacia (pobres resultados de exploración de las distintas opciones de funcionamiento y escenarios) y de eficiencia (repetición innecesaria de casos e insistencia en los menos importantes). De hecho, los propios profesionales han hecho ver que a menudo se prueba más lo que es más fácil de probar y no lo que se cree más importante.

Por último es importante destacar que no solo parece necesario la existencia de este tipo de métodos sino que, además una amplia mayoría de los profesionales participantes en la experiencia lo considerarían rentable dentro de sus organizaciones, lo cual permite despejar las dudas acerca de la posibilidad o no de asumir el sobreesfuerzo de la realización de un extenso diseño de pruebas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Programa Nacional de I+D+i del Ministerio de Educación y Ciencia y fondos FEDER, con la Red para la Promoción y Mejora de las Pruebas en Ingeniería del Software: RePRIS (TIN2005-24792-E).

Referencias

- [1] Fernández, L., Lara, P. J., Cuadrado, J.J., “Efficient software quality assurance approaches oriented to UML models in real life” en Dasso A. y Funes A. (eds.) *Verification, validation and testing in software engineering*, Idea Group, 2006, p. 341-377.
- [2] Lara, P., “Método de diseño y priorización de casos de uso a partir de especificaciones UML”, Tesis doctoral, Universidad de Alcalá, 29 de marzo de 2007.

