

Hacia una propuesta de priorización de casos de pruebas a partir de NDT

Carmen R. Cutilla, Julian A. García-García and Javier J. Gutiérrez

Grupo de investigación IWT2, Universidad de Sevilla, Sevilla, España
{carmen.ruiz, julian.garcia}@iwt2.org, javierj@us.es

Abstract. La importancia de la fase de pruebas ha ido incrementándose de manera exponencial en los últimos tiempos, llegando a considerarse hoy en día como una de las fases más importantes durante el desarrollo de software debido a los riesgos que puede suponer que el hecho de no realizar las pruebas de forma completa o incorrecta. Estos hechos quedan reflejados a la hora de definir la planificación de los proyectos, en los que la planificación se extiende en la fase de pruebas, resultando más costoso el proyecto. Esta situación conlleva la necesidad de estudiar y aplicar nuevas técnicas para ejecutar la fase de pruebas lo más completa posible reduciendo el coste de dicha fase. Entre las técnicas para conseguir este objetivo se encuentra la técnica de priorización de los casos de pruebas. Esta técnica permite generar un conjunto de casos de pruebas idóneo para validar todas las casuísticas del proyecto. En este trabajo de investigación estudiaremos una nueva línea de investigación sobre la priorización de los casos de pruebas.

Keywords: priorización casos pruebas, selección casos pruebas, casos de pruebas, ingeniería guiada por modelos

1 Introducción

La priorización de los casos de pruebas son técnicas utilizadas para disminuir los tiempos de ejecución de las pruebas [11]. Las técnicas de automatización de generación de casos de pruebas actuales son propensas a crear un número elevado de casos de pruebas, entre las que se encuentran duplicidades o casos de pruebas que se encuentran incluidos dentro de otros por estas razones, no es necesario ejecutar todo el conjunto de casos de pruebas que se han generado para validar el sistema.

Las líneas de investigación en las técnicas selección de casos de pruebas se están realizando desde hace décadas (por ejemplo, Agrawal et al. [1993], Chen et al. [1994], Harrold and Soffa [1988], Hartmann and Robson [1990], Leung and White [1990], Ostrand and Weyuker [1988], and Rothermel and Harrold [1997]). Al igual que se ha investigado en técnicas de selección de casos de pruebas, se han abierto líneas de investigación en técnicas de minimización o reducción y priorización de los casos de pruebas, demostrando sus beneficios en sus estudios de casos prácticos.

Sin embargo estas técnicas no han tenido relevancia en la práctica debido a que aún no existe ningún modelo estándar que garantice la calidad de la técnica, inclinándose en los proyectos por la automatización de la ejecución de las pruebas del conjunto completo de casos de pruebas.

La organización de este artículo se indica a continuación. En la sección 2, describiremos el caso práctico en el que hemos comprobado la justificación de una técnica o modelo que nos permita la ordenación o priorización de los casos de pruebas. En la sección 3 indicaremos los objetivos del trabajo de investigación. Las influencias de este trabajo se presentarán en la sección 4. En la sección 5 describiremos el plan de trabajo que seguiremos y por último en la sección 6 indicaremos las conclusiones recogidas.

2 Problema real

Este trabajo de investigación surge tras la participación del grupo de investigación IWT2 (Ingeniería Web y Testing Temprano) [1] en la fase de pruebas del proyecto web AQUA-WS [2]. El objetivo del proyecto AQUA-WS es el desarrollo de una plataforma web para la gestión del ciclo de las aguas de Sevilla, dividido en 4 módulos (Gestión de Obras, Gestión de Clientes, Gestión de Redes y Averías y la Oficina Virtual del Cliente), de los cuáles surgieron alrededor de 1800 requisitos funcionales complejos, compuestos por dos o más escenarios. Debido al elevado número de requisitos, el plan de pruebas generado estaba compuesto por alrededor de 7000 casos de pruebas.

Gracias a la automatización en la generación de los casos de pruebas para la elaboración del plan de pruebas [2] [4] se consiguió reducir el número de horas de trabajo en la fase de pruebas. Sin embargo al estar el proyecto AQUA-WS relacionado con funcionalidades económicas y servicios al cliente requería la necesidad de realizar una fase de pruebas completa, validando cada una de las casuísticas del proyecto, requiriendo un número elevado de horas en la fase de pruebas.

Tras el estudio de las diferentes herramientas para la ejecución automática de las pruebas como FitNesse [18], Avignon [19], JMeter [20], Selenium [21], etc, se comprobó que estas herramientas están orientadas a la ejecución de pruebas simples de navegación. En AQUA-WS el objetivo de la fase de pruebas es testear el ciclo de vida de los datos y la interrelación de éstos entre los diferentes módulos. La preparación y control de los datos con las herramientas de testeo automático resultaba más tedioso que la ejecución manual por parte de los técnicos del equipo de pruebas.

Una vez definido el equipo de pruebas para la ejecución de las pruebas, nos encontramos con el problema de tener un conjunto de 7000 casos de pruebas funcionales sin disponer un plan de ejecución de pruebas, es decir, sin tener un orden de ejecución de pruebas o sin conocer si existen casos de pruebas duplicados. ¿Qué caso de pruebas ejecutamos primero?.

3 Objetivos del trabajo

El objetivo principal de este trabajo de investigación es definir una técnica dirigida por modelos que nos permita la priorización de los casos de pruebas de manera automática o semiautomática para aplicar a la metodología NDT (Navigational Development Techniques), de modo que se pueda cubrir las necesidades indicadas en el punto anterior.

Para alcanzar este objetivo final, debemos realizar unos pasos previos de investigación, los cuales enumeramos a continuación:

1. Realizar un estudio comparativo de los métodos y algoritmos de selección y priorización de los casos de pruebas, recogiendo las ventajas e inconvenientes de cada uno de ellos.
2. Analizar de los métodos y algoritmos, con el objetivo de desarrollar un proceso de ingeniería de modelos para la priorización de los casos de pruebas.
3. Definir la automatización de la priorización de los casos de pruebas de manera automática o semiautomática.
4. Validar el proceso definido en un caso práctico con el objetivo de analizar los resultados del proceso.
5. Integrar el modelo definido en la metodología NDT.
6. Ampliar el conjunto de herramientas que ofrece actualmente NDT con la nueva herramienta de priorización de casos de pruebas.

Estos objetivos están basados en trabajos anteriores del grupo de investigación que describiremos en la siguiente sección.

4 Influencias del trabajo

La idea de este trabajo de investigación surgió durante la fase de pruebas del proyecto AQUA-WS, descrito anteriormente en este trabajo, en el que participaba el grupo de investigación IWT2 (Ingeniería Web y Testing Temprano) (Escalona, 2008) [3]. Una de las líneas de investigación del grupo es la aplicación de técnicas de testing temprano. Dentro de los recursos más importantes del grupo se encuentra la metodología NDT siendo esta metodología una de las influencias más importantes en este trabajo de investigación. Otras de las influencias a destacar es el trabajo en el que centra su estudio en la fase de pruebas y la generación de pruebas de sistemas a partir de la especificación funcional.

4.1 NDT: Navigational Development Techniques

La metodología NDT, englobada dentro del paradigma MDE (Model-Driven Engineering o Ingeniería Guiada por Modelos) fue propuesta en 2004 para la captura y análisis de requisitos en sistemas web. en su origen, esta metodología proponía una

serie de metamodelos para la fase de análisis y requisitos, y un conjunto de reglas de transformación para obtener los modelos de análisis desde el momento de requisitos. Inicialmente, NDT definía un conjunto de reglas de derivación expresadas bajo el estándar QVT (Query-View-Transformation) (OMG, 2008)[5], que permite generar los modelos de análisis a partir del modelo de requisitos. El estándar QVT define un lenguaje declarativo e imperativo propuesto por la OMG para la transformación de modelos en el contexto de la ingeniería guiada por modelos.

En la actualidad, NDT ha evolucionado hasta ofrecer soporte completo para todo el ciclo de vida del desarrollo software, siendo usado en proyectos de investigación internacionales y en el entorno empresarial. Esta metodología ha ido creciendo en otros aspectos como su enriquecimiento en temas de testing temprano o aspectos de calidad del software.

La metodología NDT está soportada por un conjunto de herramientas, NDT-Suite, apoyadas en la herramienta Enterprise Architect.

4.2 Generación de Pruebas del Sistema a Partir de la Especificación Funcional

El trabajo Generación de Pruebas del Sistema a Partir de la Especificación Funcional [Gutiérrez et al 2011] [4] es otra de las referencias de nuestro. Este trabajo también está basado en la metodología guiada por modelos NDT y los procesos de transformación (Fig. 1) del que se ha generado una herramienta para la generación de pruebas funcionales a partir de los requisitos funcionales. Esta herramienta actualmente está integrada en NDT-Suite y la cual ha sido clave en proyectos como el caso práctico AQUA-WS.

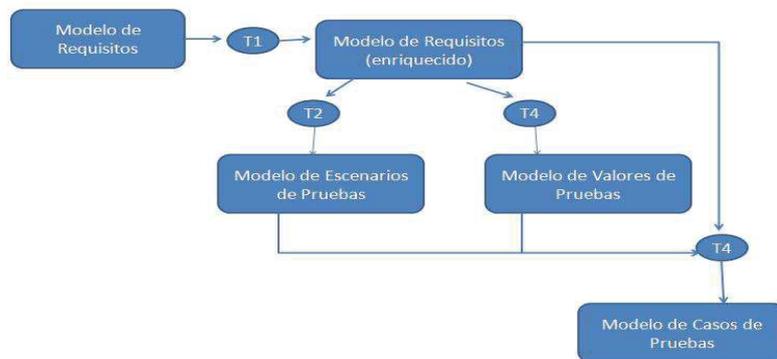


Fig. 1 Proceso de transformación Generación Casos de Pruebas

5 Plan de Trabajo

El plan de trabajo de este trabajo de investigación está constituido por una serie de hitos o tareas para alcanzar los objetivos indicados anteriormente en el apartado.

El primer hito para comenzar el trabajo es realizar un estudio comparativo sistemático de las técnicas de priorización y selección de casos de pruebas. Nuestro estudio seguiremos la guía propuesta por SEG (Software Engineering Group) [6]. Este proceso está compuesto tres actividades principales: planificación, desarrollo de la revisión y publicación de los resultados, compuesta por diferentes tareas.

En la actividad de planificación identificaremos la necesidad de realizar un estudio comparativo, definiendo el protocolo específico que debe seguir el estudio. Hasta el momento en nuestro trabajo de investigación no se han encontrado un estudio comparativo sistemático. El estudio más actualizado (S. Yoo and M. Harman, 2012) [7] realiza la comparación entre las técnicas de minimización, selección y priorización de casos de pruebas, realizando una descripción de cada una de las técnicas. En el estudio comparativo de este trabajo de investigación veremos cómo se comporta cada una de las técnicas sobre el mismo entorno. De este modo comprobamos de una manera clara y concisa las ventajas e inconvenientes de cada una de las técnicas.

Tras este estudio, analizaremos las ventajas e inconvenientes de cada una de las técnicas permitiéndonos tomarlas como referencia a la hora de la definición de la técnica guiada por modelos de nuestro objetivo principal. Una vez definido la técnica continuaremos el trabajo de investigación hasta alcanzar todos los objetivos indicados anteriormente.

6 Conclusiones

A raíz del estudio sistemático que hemos llevado a cabo, se ha comprobado que existen muchas técnicas y metodologías para la priorización de casos de pruebas pero no existe ninguna metodología estándar que garantice la calidad del proceso.

Otro factor clave es que a pesar de existir múltiples tipos de metodologías, no se están aplicando de manera general en proyectos reales o en ocasiones los testers tienen conocimiento de la existencia de éstas técnicas para agilizar su trabajo durante la ejecución de las pruebas.

Mediante la realización de este trabajo de investigación podremos definir una técnica dirigida por modelos, adaptada a la metodología NDT. Además, se pretende desarrollar una herramienta software para automatizar o semi automatizar la técnica que vamos a desarrollar. Esta herramienta se incluirá dentro de la suite de NDT: NDT-Suite.

7 Agradecimientos

Este trabajo de investigación ha sido soportado por el proyecto TEMPROS (TIN 2010-10057-C03-02) y RED CADA (TIN 2010-12313-E) del Ministerio de Ciencia e Innovación, España y el proyecto NTQ-FRAMEWORK de la Junta de Andalucía, España (TIC-5789).

8 REFERENCIAS

1. www.iwt2.org
2. C.R. Cutilla, J.A. García-García, M. Alba, M.J. Escalona, J. Ponce, L. Rodríguez, Aplicación del paradigma MDE para la generación de pruebas funcionales - Experiencia dentro del proyecto AQUA-WS (827-831)., 2011
3. Escalona, M.J., Aragón, G., NDT: A Model-Driven Approach for Web Requirements, IEEE Transactions on Software Engineering, 34(3). pp 370-390, 2008.
4. Gutiérrez, J.J, Mejías M, Escalona, M.J, , Php Generación de Pruebas del Sistema a Partir de la Especificación Funcional. 2011
5. www.omg.org
6. SEG (Software Engineering Group), Guidelines for Performing Systematic Literature Reviews in Software Engineering version 2.3. EBSE Technical Report. EBSE-2007-01. School of Computer Science, 2007.
7. S. Yoo*, M. Harman. Regression testing minimization, selection and prioritization: a survey, Volume 22, Issue 2, pages 67–120, March 2012
8. J.A.N. Lee, Xudong He, A Methodology for Test Selection, 1988
9. Harold M, Gupta R and Soffa M. “A methodology for controlling the size of a test suite” ACM Transactions on Software Engineering and Methodology, vol 2, nº3, pp.270-285,1993.
10. Harold M, Gupta R and Soffa M. “A methodology for controlling the size of a test suite” ACM Transactions on Software Engineering and Methodology, vol 2, nº3, pp.270-285,1993.
11. Polo, Macario, Garcia-Rodriguez, I.,Piattini, Mario, Priorización de casos de prueba mediante mutación, 2007
12. Jones, James Harold, Mary Jean, Test-Suite Reduction and Prioritization for Modified Condition / Decision Coverage Georgia Institute of Technology, 2003
13. Elbaum, Sebastian, Rothermel, Gregg, Kanduri, Satya, Malishevsky, Alexey G., Selecting a Cost-Effective Test Case Prioritization Technique, 2004
14. Mazeiar Salehie, Sen Li, Ladan Tahvildari, Rozita Dara, Shimin Li, Mark Moore: Prioritizing Requirements-Based Regression Test Cases: A Goal-Driven Practice. CSMR 2011: 329-332.
15. A. Srivastava and J. Thiagarajan. Effectively Prioritizing Tests in Development Environment, Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 97-106, Rome, Italy, ACM Press, 2002.
16. Jung-Min Kim and Adam Porter, A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Environments. Proceedings of the Twenty-fourth International Conference on Software Engineering. Orlando, Fl. May 2002.
17. A Multi-Objective Particle Swarm Optimization for Test Case Selection Based on Functional Requirements Coverage and Execution Efford
18. <http://www.fitnessse.org>
19. <http://sourceforge.net/projects/avignon>
20. <http://jmeter.apache.org/>
21. <http://seleniumhq.org>

Model-Based Testing in Early Software Development Phases

Silvio Cacace¹ and Tanja E.J. Vos²

¹ Dialogues Technology

{silvio.cacace}@dialoguestechnology.nl

² Universidad Politecnica de Valencia

{tvos}@pros.upv.es

1 Summary

Despite the clear advantages of using test models, drawing them is not common practice in industry. This is not in the last place because, up to date, no easy to use tool existed that enables the creation of test models and implies less maintenance when requirements change.

In this paper we will discuss a tool that does do these things, yet is very simple, light-weight, easy to learn and does not require experience and or knowledge of difficult formal methods. Basically the tool helps a tester to draw a model, by providing a drawing canvas with a range of shapes and the possibility of connecting them. Secondly, the coverage algorithms underlying the tool, will extract all the combinations of the connections between the elements, and display a minimum set of test cases that can be used for functional or acceptance testing purposes.

2 Introduction

In the eighties, Boris Beizer published his book on blackbox testing techniques [1]. The ideas in this book are simple: draw a model of the expected behaviour of the software system, and cover it. In the book, Beizer advocates that such test suite design activities should start the sooner the better. The presence of the tester in the initial development phases increases the rate of error identification, since ambiguities and unknown issues will be made clear in a very early stage and can help the business to clarify the requirements with the involvement of the stakeholders. Moreover, when testing starts early, the tester can also define the acceptance criteria in that early stage.

Using models or graphs during the early phases of testing has many identified advantages [1, 4, 6]:

- Reviewing a model is more manageable and less error-prone, than reviewing hundreds of test cases.
- Since a model is created from the requirements, it gets easier to trace the model and test cases back to the requirements.
- Models represents the tester's understanding of what the software should do In this way these models can serve as a tool for improving the communication with other stakeholders about the functionalities of the system.