

Aplicación del paradigma MDE para la generación de pruebas funcionales

Experiencia dentro del proyecto AQUA-WS

C.R. Cutilla⁽¹⁾, J.A. García-García⁽¹⁾, M.Alba⁽¹⁾, M.J. Escalona⁽²⁾, J. Ponce⁽²⁾, L. Rodríguez⁽³⁾

⁽¹⁾Fundación Fidetia. Seville. Spain.
carmen.ruiz@iwt2.org, julian.garcia@iwt2.org

⁽²⁾University of Seville. Seville. Spain
mjescalona@us.es, josep@us.es

⁽³⁾Emasesa. Seville. Spain
lrodriguez@emasesa.com

Abstract—La fase de pruebas es uno de las más críticas en Ingeniería de Software. Durante esta fase, el equipo de desarrollo tiene que asegurar que el sistema cubre las necesidades y las expectativas del cliente y de los usuarios finales. Este trabajo presenta una visión global de NDT (*Navigational Development Techniques*) que usa el paradigma MDE (*Model-Driven Engineering*, ingeniería Guiada por Modelos) para generar casos de pruebas a partir la definición de los requisitos funcionales del proyecto. Este trabajo analiza este proceso y expone cómo se ha aplicado en el proyecto AQUA-WS: un proyecto real que actualmente usa este proceso.

Keywords- *testing, testing temprano y generación pruebas funcionales*

I. INTRODUCCIÓN

Durante todo el proceso de desarrollo de un sistema software, éste debe ser comprobado para asegurar que satisface su especificación y que su funcionalidad es la esperada tanto por el cliente como por el usuario final. Con esto, se pretende mejorar la calidad del sistema desarrollado antes de su implantación.

Para cumplir este objetivo, cualquier equipo de pruebas debe disponer de un guión con las pruebas concretas a realizar, ya se ejecuten de forma manual o automáticamente. Este guión o documento se recoge en un plan de pruebas. Las cuestiones claves a analizar son cómo garantizar que las pruebas realizadas cubren completamente la funcionalidad del proyecto y cómo se garantiza que el plan de pruebas recoge todas, o al menos la mayor parte, de las casuísticas posibles.

Una de las principales técnicas para garantizar estos objetivos son las pruebas de sistema: un mecanismo para verificar que el sistema realiza el comportamiento recogido en sus requerimientos durante la fase de Requisitos del proyecto. En este sentido, algunas cuestiones clave sería investigar sobre cómo conseguir los casos de pruebas a partir de los requisitos funcionales de forma sencilla y automática.

Esta generación puede ser una forma adecuada para facilitar la ejecución de la fase de pruebas ya que se ofrecerían procesos sistemáticos para permitir la generación de pruebas

de una manera eficiente, reduciendo también el coste del proyecto.

Además, si se utilizan procedimientos automáticos se garantiza la trazabilidad del sistema final con los requisitos iniciales. En [1] se presenta una visión general para cuantificar el grado de madurez, el grado de sistematización y automatización de los trabajos existentes sobre la generación de casos de pruebas a partir de requisitos funcionales.

En la bibliografía existen múltiples planteamientos, como [2], [3] o [4] que ofrece resultados adecuados en la generación sistemática de casos de pruebas. Sin embargo, comienzan con una definición estricta de los requisitos utilizando una especificación formal, lo cual ofrece las mejores soluciones tanto para la generación de pruebas como la validación de requisitos, como se concluye en [5]. Sin embargo, son muy difíciles de aplicar en el entorno empresarial [6].

Según el estudio realizado en [1] muchas de las propuestas para generar casos de pruebas desde los requisitos funcionales presentan una tendencia general a usar el paradigma guiado por modelos adaptado a la fase de pruebas. Esta tendencia, se denomina Model-Based Testing (MBT) [7], y ha sido ampliamente utilizado en diferentes áreas de las pruebas de software.

En [8] se presenta una propuesta basada en el paradigma MBT. El procedimiento descrito en este trabajo, complementan la metodología NDT en cuanto a la generación de pruebas.

En la sección 2 de este documento se presenta una visión general de la metodología NDT, centrándose en el soporte a la fase de Pruebas.

En secciones posteriores se describe la aplicación de esta metodología sobre un proyecto real en una empresa municipal de Sevilla. En concreto, en la sección 3 se presenta una visión general de los objetivos del proyecto AQUA-WS. En la sección 4 se muestra un pequeño ejemplo de aplicación., en la sección 5 se describe cómo se han planteado las pruebas en este proyecto, y finalmente, en la sección 6 se presentan las conclusiones finales y futuros trabajos.

II. NDT- NAVIGATIONAL DEVELOPMENT TECHNIQUES

NDT Erro! A origem da referência não foi encontrada. es una propuesta metodológica incluida dentro del paradigma MDE (*Model-Driven Engineering*, ingeniería Guiada por Modelos) que se utiliza principalmente en el desarrollo de sistemas de información orientados a la WEB. NDT ofrece un proceso sistemático atendiendo a las características especiales de los entornos web.

Inicialmente, NDT estaba basado en la definición de metamodelos formales la fase de Requisitos y la fase Análisis. Además, definía un conjunto de reglas de derivación, expresadas bajo el estándar QVT (*Query-View-Transformation*) [10], con las que se generan los distintos modelos de análisis desde el modelo de requisitos. El estándar QVT es un lenguaje declarativo e imperativo propuesto por la OMG (*Object Management Group*) [11] para la transformación de modelos en el contexto de MDE.

Actualmente, NDT abarca todo el ciclo de vida software, definiendo un conjunto de metamodelos formales para cada una de las fases del ciclo de vida: desde la fase de Estudio de Viabilidad del Proyecto hasta la fase de Mantenimiento, pasando por la fase de Requisitos, la fase de Análisis, la fase de Diseño, la fase de Implementación y la fase de Pruebas. Además, establece nuevas reglas de transformación. Entre las nuevas reglas de derivación que ofrece NDT, se encuentra la que permite generar las pruebas del sistema a partir de los requisitos funcionales recogidos durante la fase de requisitos. En el contexto de UML, los requisitos funcionales equivalen a los casos de uso de UML.

Centrándonos en la fase de Pruebas, NDT la considera como una fase temprana en el ciclo de vida del software, es decir, desde el principio del proyecto y de forma paralela al resto de fases, es posible comenzar a definir los distintos casos de pruebas de sistema y articular los planes de pruebas a partir de las necesidades del sistema que han sido capturadas durante

las distintas iteraciones de la fase de Requisitos, concretamente a partir de los requisitos funcionales.

NDT propone tres modelos para la fase de Pruebas (ver figura 1): el modelo de pruebas de implantación, el modelo de pruebas de sistema y el modelo de pruebas de aceptación. Cada uno de estos modelos está descrito mediante diagramas UML de casos de uso.

De los tres modelos anteriores, el modelo de pruebas de sistema es el único que puede ser generado sistemáticamente. Como se ha adelantado anteriormente, NDT propone reglas de transformación para generar el modelo básico de pruebas de sistema a partir de los requisitos funcionales definidos en la fase de Requisitos. Esta regla de transformación está representada en la figura 1 mediante el estereotipo «*NDTTransformations*».

La regla de generación de los casos de pruebas de funcionales, depende de cómo esté definido cada requisito funcional. En NDT, los requisitos funcionales pueden ser definidos de dos formas diferentes: mediante la técnica de escenarios o mediante diagramas de actividades UML.

Si un requisito funcional está descrito mediante escenarios, serán generadas tantos casos de pruebas funcionales como escenarios tenga el requisito funcional, mientras que si está definido en base a un diagrama de actividades, serán generadas tantos casos de pruebas como caminos existan entre la actividad inicial y final del diagrama.

Las reglas de derivación no contemplan únicamente la generación de casos de pruebas de funcionales; también van construyendo un plan de pruebas a partir de las conexiones existentes entre los propios requisitos funcionales (enlaces «*extend*» e «*include*» de UML) y entre los requisitos funcionales y los requisitos de actor (enlaces «*use*» de UML). Los requisitos de actor también son recopilados durante la fase de Requisitos.

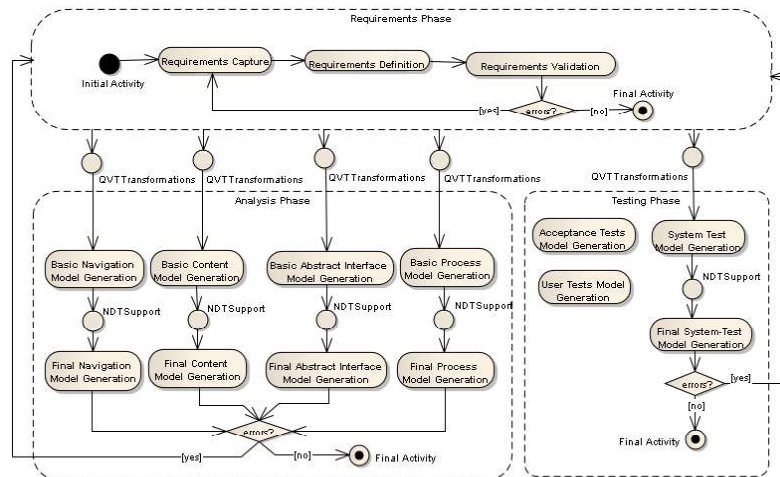


FIGURA 1: GENERACIÓN DE LOS MODELOS DE LA FASE DE ANÁLISIS Y PRUEBAS DESDE LA FASE DE REQUISITOS.

Una vez que ha sido generado el modelo básico de pruebas, el equipo de analistas pueden realizar transformaciones controladas con el objetivo de completarlo y enriquecerlo para obtener el modelo final. Este paso no es automático y requiere la experiencia del analista. Estas transformaciones están representadas en la figura 1 mediante el estereotipo «NDTSupport».

La aplicación de MDE y, en particular, la aplicación de las transformaciones de los modelos pueden llegar a ser monótona y suponer un coste elevado si no se dispone de herramientas software que automaticen el proceso.

Para satisfacer esta necesidad, NDT ha definido un conjunto de herramientas libres e implementadas en Java, para dar soporte al ciclo de vida de un proyecto software. Este conjunto de herramientas se distribuyen bajo el nombre NDT-Suite [12].

III. PROYECTO AQUA-WS

Emasesa **Erro! A origem da referência não foi encontrada.** es una compañía local encargada de llevar a cabo la gestión general del ciclo del agua en la ciudad de Sevilla y en algunas de sus localidades limítrofes. Esta empresa tiene como objetivos proporcionar y garantizar la calidad del suministro de agua a los ciudadanos de estas localidades, resolviendo cualquier posible problema que pueda surgir en el mismo y controlando la correcta utilización del agua.

Actualmente, para garantizar estos objetivos, Emasesa dispone de un conjunto de sistemas informáticos independientes y desarrollados con tecnologías diferentes:

- AQUA-SiC, sistema de gestión de clientes. Este sistema es una aplicación local.
- AQUA-ReD, sistema de gestión de redes basado en tecnologías web.
- AQUA-SigO, sistema de gestión de obras y proyectos basado en tecnologías web.

El proyecto AQUA-WS (AQUA-WebServices) surge a raíz de la necesidad de unificar todos los sistemas actuales en un único sistema, el núcleo AQUA, utilizando además la misma tecnología. El nuevo sistema está orientado a la web y a su desarrollo se ha utilizado Java J2EE.

Además de lo mencionado anteriormente, el proyecto AQUA-WS tiene como principales objetivos diseñar un sistema integrado para la toma de decisiones (DSS, *Decision Support System*), garantizar de forma eficaz la comunicación entre el Núcleo AQUA con otros sistemas transversales utilizados en Emasesa a través de un bus de integración y por último implementar una interfaz de usuario amigable en proporcionar una conexión adecuada tanto para usuarios como empleados de Emasesa. Este último objetivo es lo que se conoce como módulo FrontEnd.

La figura 2 presenta un diagrama con la arquitectura del proyecto AQUA-WS.

El núcleo AQUA, el módulo FrontEnd, el sistema DSS y la comunicación con sistemas transversales están basados en

el uso de SAP ¹ ERP **Erro! A origem da referência não foi encontrada.** (*SAP Enterprise Resource Planning*), SAP BW **Erro! A origem da referência não foi encontrada.** (*SAP Netweaver Business Warehouse*), GIS (*Geographical Information System*) sobre un sistema de gestión documental (*Documental Management*).

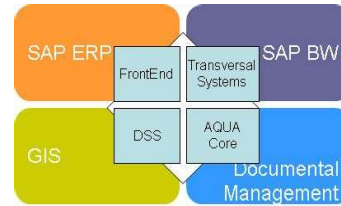


FIGURA 2: ARQUITECTURA AQUA-WS

En la tabla 1 recogemos datos cuantitativos que dan una idea de la complejidad del proyecto.

TABLA 1. TOTAL DE REQUISITOS FUNCIONALES EN AQUA-WS

Modulo	Nº de Requisitos Funcionales
SIGO	436
RED	334
SIC	730
Bus de Integración	40
FrontEnd	268
TOTAL	1.808

Del total de requisitos funcionales, el 70% son requisitos funcionales de complejidad alta (con más de 15 pasos por escenario), el 20% de complejidad media (entre 5 y 14 pasos por escenario) y sólo el 10% se consideran de complejidad baja (menos de 5 pasos por escenario).

Además, existe un alto grado de interrelación entre los módulos, lo que incrementa enormemente el grado de complejidad del proyecto AQUA-WS.

IV. EJEMPLO DE APLICACIÓN DE NDT-DRIVER

En esta sección se ilustra a través de un pequeño ejemplo la aplicación de NDT-Driver en la generación de los casos de pruebas funcionales a partir de un requisito funcional.

La Figura 2, muestra el diagrama de actividades de un requisito funcional recogido en la fase de requisitos del proyecto AQUA-WS. Este requisito funcional describe cómo se asocia una finca a un cliente a través de su contrato de suministro de agua.

Como se indicó en la sección 2, se generan tantos casos de pruebas como caminos existan entre la actividad inicial y la actividad final: un caso de prueba en el que el cliente está registrado en el sistema y la finca está registrada en el

¹ Systems, Applications and Products in Data Processing

sistema, por lo que los pasos del caso de prueba consistiría en buscar el cliente y completar el formulario, PS01; un segundo caso de prueba en el que el cliente no está registrado, que registraremos previamente pero sí está registrada la finca a realizar el contrato, la prueba consistiría en buscar el cliente, registrarlo y verificar que existe la finca y completar finalmente el formulario de contrato, PS02; y una tercer caso de prueba donde no están registrados el cliente y la finca, que se realizaría previamente y se completará el formulario de alta contrato, PS03.

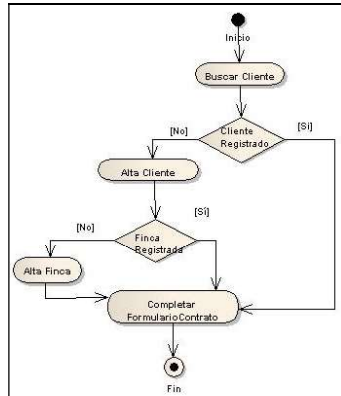


Figura 2: Requisito Funcional para el ejemplo

En la Figura 3 se muestra de izquierda a derecha los casos de pruebas descritos anteriormente, PS01, PS02 y PS03.

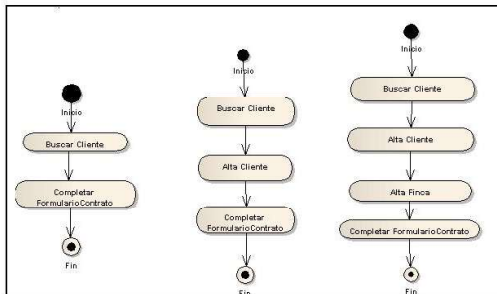


Figura 3: Casos de Pruebas para el ejemplo

V. FASE PRUEBAS DE AQUA-WS

La importancia de la fase de pruebas está contemplada en la planificación del proyecto AQUA-WS, destinando el 40% de la duración del proyecto en los diferentes tipos de pruebas.

A esta fase se le ha dotado de un grupo de técnicos liderado por miembros de la Oficina Técnica de Calidad. El resto del equipo está formado por miembros de la empresa

de mantenimiento de los distintos sistemas actuales, y miembros de las empresas que están llevando a cabo el desarrollo del sistema AQUA-WS.

El principal objetivo del equipo de pruebas es realizar las pruebas de aceptación en un entorno de preproducción para comprobar que toda la funcionalidad de los sistemas actuales está recogida en el nuevo sistema.

Las pruebas de aceptación se obtienen como subconjunto de las pruebas de sistema generadas por NDT-Driver (herramienta descrita en la sección 2) y recogidas en el Plan de Pruebas.

El número de pruebas que recogen los distintos planes de prueba en este proyecto son muy elevadas. Tomando como media cuatro escenarios por cada requisito funcional, y tengan cuenta que existen 1808 requisitos funcionales, en un proyecto de esta envergadura se generan del orden de 7000 pruebas funcionales.

Realizar de forma manual el plan de pruebas completo, a una media de 20 minutos en documentar una prueba funcional, en un proyecto de 7000 pruebas supone un coste del orden de 2300 horas de trabajo.

Con el uso de la herramienta NDT-Driver, la obtención de la primera versión del plan de pruebas es casi inmediata, únicamente se necesitaría una aproximación de 4 minutos para incluir el juego de las dos de cada prueba. Lo que supondría unas 500 horas de trabajo empleadas en esta tarea. Analizando estos datos se comprueba que gracias a NDT-Driver se consigue ahorrar 1800 horas de trabajo.

Una vez generado el plan de prueba, comienza la fase de ejecución de pruebas. Actualmente se está trabajando en el desarrollo de aplicaciones para la automatización de todo tipo de pruebas, como son:

- Pruebas aceptación: FitNesse, Avignon
- Pruebas rendimiento: JMeter, OpenSTA, WebLoad, Grinder
- Pruebas funcionales: Selenium IDE, HTTPUnit, BadBoy, SAHI.

Tras el estudio de las diferentes herramientas para la ejecución automática de pruebas funcionales, se analizaron los objetivos y funcionamiento de cada una de ellas.

Todas estas herramientas se basan en pruebas de navegación simples y orientadas a realizar una repetición de dichas pruebas. El objetivo de las pruebas funcionales en AQUA-WS es testear el ciclo completo de los datos y validar la interrelación de los datos con los diferentes sistemas.

Tras el análisis de herramientas, se descartaron todas ellas ya que no estaban orientadas al objetivo de AQUA-WS y resultaba ser más tedioso la preparación y control de las pruebas automáticas que la realización individual por parte de los técnicos del equipo de pruebas.

Las pruebas funcionales son los más importantes en el proyecto AQUA-WS. Su realización se estructura en dos

ciclos de pruebas que el equipo de pruebas lleva a cabo. En el primer ciclo, se realiza un gran número de pruebas con el objetivo de validar el mayor número de casuísticas posible. Estas pruebas se han llevado siguiendo el plan de pruebas, que se ha generado mediante la herramienta NDT-Driver (descrito en la sección 2). A medida que se va realizando las pruebas, se van detectando incidencias gestionándolas mediante un proceso diseñado por la Oficina de Calidad y el jefe de proyecto. Este procedimiento se basa en diferenciar los diferentes tipos de incidencias y tipificar su grado gravedad.

Una vez que el equipo de desarrollo ha resuelto todas las incidencias, comienza el segundo ciclo de pruebas. Las pruebas recogidas en el plan de pruebas son ejecutadas para la verificación y validación de que todas las incidencias reportadas han sido resueltas correctamente y no se ha generado ninguna incidencia por efecto colateral.

VI. CONCLUSIONES Y FUTUROS TRABAJOS

En Ingeniería del Software y en particular en Ingeniería Web, es importante llevar a cabo completa y exhaustivamente la fase de Pruebas dentro del ciclo de vida del proyecto con el objetivo de asegurar la calidad del producto desarrollado y para asegurar que cubre las necesidades y expectativas de clientes y usuarios finales.

Para lograr estos objetivos la fase de pruebas debe ser planificada con el tiempo suficiente para su consecución y además debe ser dotada de los recursos necesarios, tanto a nivel técnico como humano.

Este trabajo presenta cómo la metodología NDT proporciona soporte a la fase de Pruebas y además presenta su suite de herramientas software, y en particular su herramienta NDT-Driver, la cual permite disminuir el tiempo invertido en la generación de los casos de prueba.

Además, en este trabajo se ha presentado los resultados y conclusiones de la aplicación de esta metodología y de sus herramientas, en un proyecto real de gran envergadura: el proyecto AQUA-WS.

El proyecto AQUA-WS ha abierto una importante línea de investigación a explorar. El uso de técnicas automáticas para la generación de casos de pruebas funcionales del sistema a partir de los requisitos funcionales capturados durante la fase de Requisitos del proyecto, ha permitido disminuir el tiempo que el equipo de analistas ha tenido que invertir en desarrollar los distintos planes de pruebas.

Como líneas futuras de investigación, proponemos el uso de MBT para mejorar la fase de pruebas en la metodología NDT. Actualmente, NDT proporciona sólo la generación de casos de prueba. En el futuro, intentaremos progresar en esto no sólo para mejorar el procedimiento de generación automático de casos de prueba sino también para generar el juego de datos con los que probar.

Además, proponemos también como líneas futuras de investigación cómo crear paneles de control con el objetivo de automatizar la gestión de la fase de pruebas y evitar que

se realice de forma manual que es cómo transcurre normalmente en la actualidad.

AGRADECIMIENTO

En este trabajo de investigación ha sido soportado por el proyecto QSimTest (TIN2007-67843-C06_03), por el proyecto Tempros del (TIN2010-20057-C03-02), España y por la red Nacional CaSA (TIN2010-12312-E) del Ministerio de Ciencia y Educación.

REFERENCIAS

- [1] M.J. Escalona, J.J. Gutierrez, M. Mejías, G. Aragón, I. Ramos, J. Torres. An Overview about Test Generation from Functional Requirements.
- [2] Stocks, P., Carrington, D. A framework for specification-based testing. *IEEE Transaction on Software Engineering*. Vol. 22. N. 11. 1996.
- [3] Liu, S., Yuting, C. A relation-based method combining functional and structural testing for test case generation. *Journal of Systems and Software*. Vol. 81. 2. pp. 234-248. 2008.
- [4] TerMaat, P. Adventures in Automated Testing! The Software Testing and quality engineering magazine. May/June 2001.
- [5] Ryser, J., Glinz, M. A Practical Approach to Validating and Testing Software Systems Using Scenarios Quality. *Week Europe QWE'99 in Brussels, Institut für Informatik, Universität Zürich*. 1999.
- [6] E. Insfrán, O. Pastor, R. Wieringa. "Requirements Engineering-Based Conceptual Modelling". *Requirements Engineering Journal*, Vol 7 (1). 2002.
- [7] Legeard, B. Model-based Testing: Next Generation Functional Software Testing. *Dagstuhl Seminar Proceedings 10111. Practical Software Testing: Tool Automation and Human Factors*. 2010.
- [8] Gutiérrez, J.J., Escalona, M.J., Mejías, M., Torres, J., Torres-Zenteno, A.H. A Case Study for generating test
- [9] Escalona, M.J., Aragón, G., 2008. NDT: A Model-Driven Approach for Web Requirements, *IEEE Transactions on Software Engineering*, 34(3). pp 370-390.
- [10] OMG, 2008. Documents Associated with Meta Object Facility (MOF) 2.0 Query/View/Transformation. <http://www.omg.org/spec/QVT/1.0/>
- [11] OMG, 2005. Unified Modelling Language: Super structure. Specification. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>
- [12] NDT-Suite. Available in www.iwt2.org/ndt
- [13] Emasesa. Empresa Municipal de Aguas de Sevilla. <http://www.aguasdesevilla.com>
- [14] SAP ERP: www.sap.com/ERP
- [15] SAP BW: <http://www.sap.com/solutions/benchmark/bw.epx>
- [16] Hartman A., Nagin, K. 2003. Model Driven Testing-AGEDIS Architecture Interfaces and Tools. In 1st European Conference on Model Driven Software Engineering, Nuremberg, Germany, pp 1-11.