

Gestión de pruebas en CALIPSOneo: Una solución MDE

A. Salido¹, J. A. García-García¹, J. Ponce¹, J. J. Gutierrez¹, M. Oliva²

alberto.salido@iwt2.org, julian.garcia@iwt2.org, josepg@us.es, javierj@us.es, manuel.oliva@military.airbus.com

¹Software Engineering and Early Testing Group, Universidad de Sevilla
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
AIRBUS
Av. García Morato s/n, 41011, Sevilla, Spain

Resumen: La fase de pruebas siempre es una de las más importantes en el desarrollo de software. Sin embargo, cuando los proyectos sufren algún retardo en las fases de desarrollo, este retraso suele repercutirse de manera negativa en la fase de pruebas. La ingeniería guiada por modelos (MDE-Model Driven Engineering) ofrece una solución para paliar el coste de las pruebas sin repercutir en la calidad de la ejecución de las mismas. Esto se enfoca dentro del área que se suele denominar Pruebas guiadas por Modelos (MDT-Model Driven Testing). En este trabajo, se presenta una visión práctica de una propuesta metodológica basada en MDT y analiza las repercusiones que ha tenido en este trabajo empírico. Para ello, se utilizará el ejemplo del proyecto CALIPSOneo. Un proyecto realizado en colaboración con Airbus que, además, supone un reto en este entorno por la propia naturaleza del proyecto enmarcada dentro de los entornos PLM (Product Lifecycle Management).

Palabras claves: Model-Driven Testing, Model-Driven Engineering, Product Lifecycle Management.

Abstract: The testing phase is always, one of the most important in the development of software. The literature is covered of research results and references that reflect this. However, this phase is often approached in a way less structured and less strategy in practice. Thus, when projects experience some delay in the development phases, this delay is often passed negatively in the testing phase. The Model Driven Engineering (MDE) can offer a solution to alleviate the cost of testing without compromising the quality of its execution. In this context, this approach is called Model-Driven Testing (MDT). In this paper, we present a practical overview of a MDT-based methodology and we analyze the impact that MDT has had on the empirical work: the CALIPSOneo project. This project is done in collaboration with Airbus, which also poses a challenge in this environment because of the nature of the project framed within the PLM environment (Product Lifecycle Management).

Keywords: Model-Driven Testing, Model-Driven Engineering, Product Lifecycle Management.

1 Introducción

La fase de pruebas es una fase fundamental, sin embargo, por regla general, no se suele contar con una metodología de trabajo predefinida que garantice los resultados de dicha fase.

Por la experiencia práctica de los autores, en muchos casos, la fase de pruebas es la que se ve recortada cuando se producen retardos o retrasos en el proyecto, reduciendo el alcance de las mismas.

En el caso concreto de las pruebas funcionales, que deben estar orientadas principalmente a los usuarios, el problema además es que suelen ser diseñadas por personas que no participaron en la elicitación de requisitos. Por ejemplo, los propios programadores. Esto hace que, en muchas ocasiones, se prueben aspectos funcionales que, si bien “funcionan bien” no se corresponden con las necesidades que inicialmente fueron definidas.

En la búsqueda de solventar estos aspectos, una de las iniciativas que se aplica en el contexto de investigación es la prueba guiada por modelos [Bertolino2005]. Este paradigma puede ayudar a la generación sistemática e independiente del proceso de desarrollo y puede ayudar a reducir la generación de las pruebas.

Este trabajo presenta esta idea en el contexto de la metodología NDT (*Navigational Development Techniques*) [Escalona2008] y la analiza en su aplicación

en un contexto muy concreto, el proyecto CALIPSOneo. Este proyecto presenta un entorno especial de evaluación por estar definido en el contexto de las soluciones PLM (Product Lifecycle Management) [Stark2011].

El trabajo se estructura analizando el contexto de la ingeniería guiada por pruebas en la Sección 2. Tras esto, en la Sección 3, se presenta tanto una visión global del proyecto CALIPSOneo como el contexto metodológico en el cual ha transcurrido del proyecto. La Sección 4 analiza la aplicación concreta de nuestra solución al entorno concreto de CALIPSOneo. El artículo finaliza con las conclusiones y trabajos futuros.

2 Model-Driven Testing

El diseño de pruebas guiadas por modelos es algo que se va estandarizando cada día más a la hora de realizar todo el proceso de automatización de pruebas de un determinado desarrollo software.

Para poder llevar a cabo dicha forma de realizar las pruebas, una de las opciones es basarse en el estándar UTP (en) [U2TP] lo que nos proporciona una metodología (introducida por Dai) [ZRDai2004]. Esta propuesta propone usar U2TP con el fin de transformar un modelo de sistema UML en modelos de pruebas (todo ello gracias a reglas de transformación Query/View/Transformation – QVT) [OMG2013].

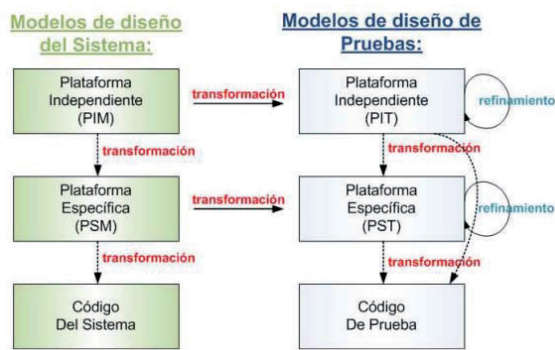


Figura 1. Propuesta usada en el perfil U2TP.

Como bien se puede observar en la figura anterior, los modelos de diseño del sistema independientes de la plataforma (PIM) pueden ser transformados en modelos de diseño del sistema específicos de la plataforma (PSM) teniendo en otro paso de la transformación, el código del sistema que pueda derivar del PSM.

U2TP proporciona conceptos para desarrollar especificaciones y modelos de pruebas para pruebas de caja negra [Beizer1995]. El profile introduce cuatro grupos de conceptos lógicos que cubren los siguientes aspectos: arquitectura de las pruebas, comportamiento de la prueba, datos de prueba y tiempo. Juntos, estos conceptos definen un lenguaje de modelado para visualizar, especificar, analizar, construir y documentar un sistema de pruebas.

Existen más estudios sobre estas metodologías que aplican ciertas ampliaciones del metamodelo UML para agregar información requerida para el oráculo de las pruebas en los diagramas de secuencia: puesto que un diagrama de secuencia se corresponde con un escenario relevante que debe ser probado, el diagramase anota de forma que incluya información sobre el resultado esperado y el estado inicial del mismo [BeatrizP2008].

Pero el concepto de Pruebas guiadas por modelos ha sido aplicado en muchos más contextos. En el trabajo [Escalona2012] se hace un recorrido comparativo por cómo algunas propuestas proponen la generación sistemática de pruebas software, pruebas funcionales en este caso. Varias de estas propuestas parten de los modelos de requerimientos para llegar a los modelos de pruebas funcionales.

Dentro de las propuestas que se presentan aquí, encontramos la propuesta [Gutierrez2012] que se basa totalmente en la estructura mostrada en la Figura 1. En ella, se parte de un modelo de requerimientos funcionales, que se define como una instancia de un metamodelo definido en la propuesta, y, a través de transformaciones QVT, se definen las pruebas funcionales. Esta va a ser la partida de nuestro trabajo que ha sido adaptada al proyecto CALIPSONeo.

3 CALIPSONeo

A continuación se presenta una visión general del proyecto CALIPSONeo y de cómo se ha abordado su fase de pruebas.

3.1 Visión global del proyecto

CALIPSONeo (advanCed Aeronautical soLutions using Plm procesSes & tOols) es un proyecto liderado por Airbus que tiene como principal objetivo la definición de una metodología de trabajo que permita a los ingenieros definir, simular, optimizar y validar los procesos de montaje aeronáuticos en un entorno virtual 3D antes de su ejecución real en una línea de montaje. Todo ello a través de un proceso integral de recolección de requisitos y de la customización y el uso de software PLM existente en el mercado.

A través de los años, se realizaron varios análisis en PLM que concluyeron con un grupo de beneficios para Airbus Military [Mas2012]. A partir de este conocimiento, el equipo del proyecto CALIPSONeo ha recopilado los distintos requisitos del proyecto, el plan de gestión de requisitos, y la matriz de trazabilidad de requisitos que satisface las necesidades del negocio PLM. Por tanto, CALIPSONeo cubre el diseño de una nueva metodología PLM para ajustarse a una solución PLM colaborativa y el desarrollo de la solución software que satisface ese concepto.

CALIPSONeo se divide en tres subproyectos individuales con el fin de gestionar, de forma eficaz, los trabajos necesarios para su realización. Esto permite llevar a cabo una gestión más eficiente del alcance del proyecto a medida que los distintos equipos de trabajo culminan las tareas necesarias para completar el proyecto. Los tres subproyectos son: MARS (*autoMated shop-floOR documentation updating System*), PROTEUS (*PROcess sTructure gEneration and Use*) and ELARA (*gEnerALization to assembly oriented authoring Augmented ReALity*).

El objetivo de ELARA es desarrollar una solución Software que permita, a los operarios de la línea de montaje, visualizar en 3D las tareas de montaje que deben llevar a cabo. Todo esto por medio de dispositivos móviles con software de realidad aumentada. La información mostrada vendrá procedente de la información en 3D contenida en la iDMU (industrial Digital Mock Up) [Mas 2013] y la información procedente del sistema MARS. El resultado final será un prototipo industrial para ser utilizado en el proceso de montaje de los FanCowls de los motores del Airbus A320neo.

En cuanto al subproyecto **PROTEUS**, éste es el responsable de la definición de una estructura de datos que permitan interrelacionar productos, procesos y recursos con el objetivo de crear la estructura de información iDMU.

Finalmente, MARS será el sistema responsable de la explotación de iDMU generada por el proyecto PROTEUS para obtener automáticamente la documentación que necesita el operario de la línea de montaje para ensamblar la pieza correspondiente del avión. Tanto MARS como PROTEUS son proyectos metodológicos, en los cuales se necesita desarrollar un modo de trabajo. Estos proyectos cuentan con el desarrollo de algunos scripts de código, pero el foco principal es la creación de una metodología de trabajo en

la cual se explique al usuario final cómo realizar un tipo específico de tarea.

Aunque cada uno de los proyectos descritos anteriormente tiene bien definidos su alcance y objetivos, es necesario recalcar que, durante su desarrollo, se está requiriendo un alto grado de integración para garantizar que toda información que discurre entre los sistemas sea correcta y coherente en todo momento.

3.2 Entorno metodológico

Para la consecución de los objetivos establecidos en los tres subproyectos de CALIPSOneo se ha optado por utilizar la metodología NDT [Escalona2008]. Esta metodología está englobada dentro del paradigma MDE y concretamente, para las pruebas, usa una extensión basada totalmente en la propuesta definida por [Gutierrez2012], que hemos definido en el apartado 2 de este trabajo.

Actualmente, NDT contempla y define un conjunto de metamodelos para cada una de las fases del ciclo de vida de desarrollo de software: Estudio de Viabilidad, Requisitos, Análisis, Diseño, Implementación, pruebas y Mantenimiento. Además, establece un conjunto de reglas de transformación QVT con las que es posible generar de manera sistemática unos modelos a partir de otros, redundando así en un menor coste asociado a la especificación del desarrollo de software.

Sin embargo, para que toda la potencia de NDT sea práctica y de utilidad en entornos empresariales es necesario disponer de herramientas que automaticen todo el proceso. Para ello, NDT cuenta con un conjunto de herramientas distribuidas bajo el nombre de NDT-Suite¹¹ [García-García2012]. Las principales herramientas que componen esta suite son: (i) NDT-Profile, que define perfiles UML en Enterprise Architect¹² para cada uno de los metamodelos de NDT; (ii) NDT-Quality [García-García2013], que permite medir de forma automática la calidad en el uso de la metodología en cada una de las fases del ciclo de vida y comprobar la correcta trazabilidad de las reglas MDE de NDT; y (iii) NDT-Driver [García-García2011], que permite la aplicación automática de todas las reglas QVT definidas en NDT.

En los últimos años, NDT ha vuelto a evolucionar y actualmente ofrece un marco de trabajo eficaz y global para la aplicación práctica de NDT, denominado NDTQ-Framework¹³. Este marco de trabajo contempla un conjunto de procesos enmarcados dentro de procesos de desarrollo de software, procesos de calidad, procesos de pruebas, de gestión y procesos de seguridad. NDTQ-Framework está basado en diferentes modelos de referencia, como CMMi (*Capability Maturity Model Integration*) [Chrissie2011] e ITIL (*Information Technology Infrastructure Library*) [Jong2008], y su aplicación en proyectos reales está certificado bajo diferentes estándares, como ISO 27001, ISO 9001:2008, UNE EN 16602 y ISO 14000.

¹¹ Disponible en <http://www.iwt2.org>

¹² Disponible en www.sparxsystems.com.au

¹³ Este artículo presenta NDTQ-Framework en detalle, pero puede consultar más información en <http://www.iwt2.org>.

Debido a las peculiaridades del proyecto CALIPSOneo, no ha sido posible aplicar, de forma directa, la metodología NDT, por lo que fue necesario realizar una adaptación de la misma a este contexto obteniendo así muy buenos resultados. Las adaptaciones han sido las siguientes.

Por una parte, y, como se ha mencionado anteriormente, CALIPSOneo es en realidad un proyecto formado por tres subproyectos. Este aspecto ha motivado la adaptación de las reglas de transformación MDE de NDT para soportar la integración y coordinación entre los tres subproyectos.

Por otra parte, debido a que CALIPSOneo se ha desarrollado dentro de un contexto PLM, se ha generado muy poco software. De hecho, en este proyecto tanto, el modelo de requisitos como el de análisis, adquiere un valor central y estratégico en el proyecto, ya que son el punto angular que garantiza la calidad de los resultados. En este sentido, NDT-Profile proporciona un entorno ágil de trabajo para los equipos. Por otra parte, debido al elevado volumen de productos documentales generados en el proyecto, fue necesario utilizar repositorio documentales basados en herramientas ECM¹⁴. Además, la propia metodología contempla la trazabilidad entre estos productos documentales y los requisitos.

4 Fase de pruebas y resultados

Para la fase de pruebas en CALIPSOneo se ha utilizado la herramienta NDT-Driver una de las herramientas de soporte de NDT y con la que ha sido posible generar, de forma sistemática y automática, el conjunto de pruebas de sistema a partir de los requisitos funcionales de cada uno de los tres subproyectos de CALIPSOneo.

El proceso de ejecución propiamente dicho de las pruebas ha sido adaptado según la tipología del proyecto, es decir, como se comentó anteriormente, tanto PROTEUS como MARS tienen un carácter sumamente metodológico, mientras que ELARA es fundamentalmente un proyecto de desarrollo de software.

En el caso de los **proyectos metodológicos**, el proceso de validación se ha realizado siguiendo tres fases. La Figura 2 muestra, de forma esquematizada, todo el proceso metodológico de validación. Este proceso está basado en tres fases: fase de revisión y validación según las normas establecidas por NDT, fase preliminar y fase exhaustiva.

En la primera fase, es decir, la fase de validación, el equipo de pruebas realiza tareas propias de una oficina técnica de calidad revisando y validando los planes de pruebas (utilizando NDT-Quality) desarrollados por los proveedores software del proyecto. Estos planes de pruebas son generados en primera instancia en versión borrador (*draft*) y almacenados en el repositorio documental del proyecto. Estas tareas de calidad nos han permitido trabajar de forma paralela y colaborativa entre el equipo de pruebas y el equipo de desarrollo, con el objetivo de disponer de unos planes de pruebas exhaustivos.

¹⁴ ECM es el acrónimo de *Enterprise Contents Management* y la solución utilizada dentro del contexto de CALIPSOneo fue Alfresco. Usted puede encontrar más información sobre Alfresco en su sitio web.

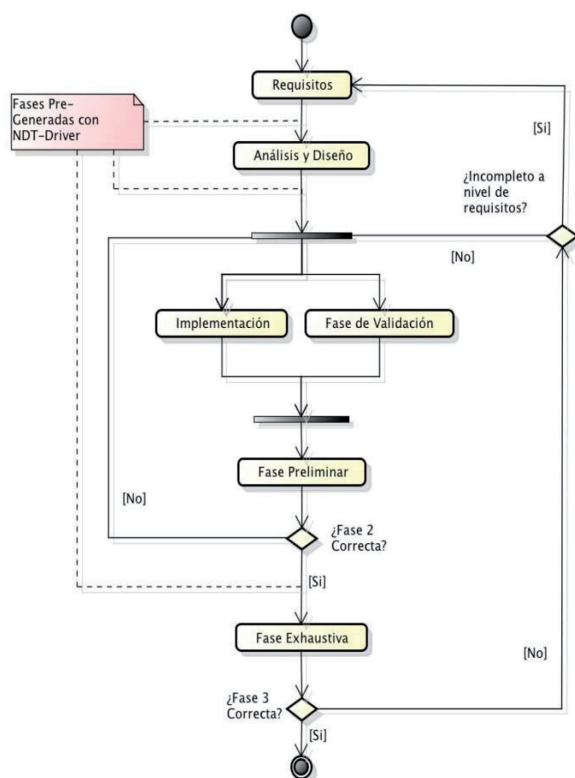


Figura 2. Diagrama de desarrollo de las pruebas.

Una vez desarrollado y validado las tres cuartas partes del plan de pruebas, el equipo de pruebas comienza la siguiente fase, la fase de pruebas preliminares, en las que el objetivo principal es detectar el mayor número posible de incidencias. En esta fase, el equipo de pruebas tiene como objetivo reproducir cada una de las actividades definidas en el documento metodológico desarrollado en el contexto del subproyecto (MARS o PROTEUS). Por ejemplo, el proyecto MARS define un proceso metodológico para generar instrucciones de trabajo (*Work Instructions*) tanto en formato PDF (para ser impresa en papel o visualizada en un dispositivo móvil tipo Tablet) como en 3DXML para ser visualizadas en un explorador Web. Estas instrucciones de trabajo agilizan el trabajo que debe realizar cada operario, ya que proporciona la información de una manera estructurada, dinámica y concisa.

Finalmente, cuando el plan de pruebas se encuentra en un estado de validación del 90% o superior, el equipo de pruebas lleva a cabo una fase exhaustiva de pruebas en las que el objetivo es realizar la aceptación final de cada una de las propuestas metodológicas definidas en el proyecto MARS y PROTEUS.

Después de esta última fase del ciclo de pruebas en CALIPSOneo, el equipo de pruebas reporta un informe con los resultados de las mismas, indicando su tipología, juego de datos utilizado y una descripción detallada del error. Como valor añadido, durante esta fase el equipo de pruebas ha sido capaz de proponer mejoras de optimización sobre los procesos metodológicos definidos.

La aplicación directa de estas fases de pruebas garantiza una validación constante de los documentos según van

siendo éstos realizados y además contribuye a mejorar la calidad de los entregables. De esta forma, se acelera el proceso de pruebas, la cual, a pesar de su reconocida necesidad, sigue siendo, en la práctica, la fase que más recortes de recursos sufre cuando los proyectos van con retraso

Por otra parte, en el caso del subproyecto ELARA, la fase de pruebas, en su conjunto, se ha abordado desde una perspectiva distinta, pero manteniendo las bases metodológicas de NDT. En este caso, cada caso de prueba se ha definido teniendo en cuenta diferentes variables y situaciones en las cuales el software debe garantizar un funcionamiento correcto. Por ejemplo, el proyecto ELARA consta de un módulo de ajuste (*Matching*), el cual, utilizando la cámara del dispositivo (*Tablet PC*) y un modelo virtual superpuesto, sincroniza la imagen real con la virtual, ayudando al operario a realizar sus operaciones de montaje de una forma dinámica.

Uno de los problemas que más se ha presentado durante la fase de pruebas de este proyecto ha sido, la poca madurez de los diagramas realizados en las fases de análisis y diseño, teniendo como consecuencia, unos casos de prueba no lo suficientemente exhaustivos como para cubrir las necesidades de requisito funcional. Debido a esto, se han tenido que modificar los requisitos del proyecto, y autogenerarlos de nuevo utilizando NDT-Driver.

En resumen, se ha verificado la eficiencia de NDT-Suite con proyectos tecnológicos, no solo en ámbito Software, sino también en el desarrollo de metodologías de trabajo agilizando la especificación sobre el qué se quiere obtener y ayudando a su desarrollo y validación.

5 Conclusiones y Trabajos Futuros

En este trabajo, se ha presentado un análisis del trabajo realizado en el contexto del proyecto CALIPSOneo aplicando la ingeniería guiada por modelos para el trabajo con las pruebas funcionales.

El trabajo ha permitido validar los resultados de investigación en un contexto real. NDT ha sido una propuesta que se ha aplicado en diferentes contextos reales, pero el contexto PLM tiene una serie de características especiales que hacían atractiva la validación en este ámbito.

Del trabajo, podemos concluir que la propuesta ofertada presenta una solución válida, pero también que aún requiere de un gran trabajo manual por parte de los gestores y el equipo de prueba.

CALIPSOneo ha permitido identificar un conjunto de requerimientos para poder mejorar la propuesta. Por un lado, nuevamente hemos vuelto a ratificar la necesidad de disponer de unos requerimientos bien definidos para garantizar una buena definición de las pruebas. Esto nos ha abierto una nueva línea para investigar en alguna de las opciones de NDT-Suite. Ésta incluye una herramienta, denominada NDT-Quality, que permite validar la calidad de todos los artefactos que se definen en NDT. Nuestra idea es investigar si podemos enriquecerla para ver si podemos sistematizar, o mejor aún, automatizar la

revisión funcional de los requerimientos desde las primeras fases del ciclo de vida.

Otros aspectos a mejorar es la propia especificidad de los contextos PLM. De esta manera, se intentarán mejorar las herramientas para la gestión de las pruebas.

Referencias bibliográficas

- [Mas2012] Mas, F., Rios, J., Menendez, J.L., Gomez, A., "A process-oriented approach to modeling the conceptual design of aircraft assembly lines", *International Journal of Advanced Manufacturing Technology* 2012, Vol. 62, 2012.
- [Mas2013] Mas, F., Gomez, A., Menendez, J.L., Rios, J., "Proposal for the conceptual design of aeronautical final assembly lines based on the iDMU concept". A. Bernard, L. Rivest, and D. Dutta (Eds.): *PLM 2013*, IFIP AICT 409. IFIP International Federation for Information Processing 2013.
- [OMG2013] Query/View/Transformation. www.omg.org/spec/QVT/1.1/. Release 1.1. 2011. Accessed 05/2013.
- [Escalona2008] Escalona, M.J., Aragón, G. "NDT: A Model-Driven Approach for Web requirements", *IEEE Transactions on Software Engineering*. Vol. 34. Nº 3. pp 370-390, 2008.
- [García-García2012] García-García, J.A., Ortega, M.A., García-Borgoñón, L., Escalona, M.J.: "NDT-Suite: A Model-Based Suite for the Application of NDT". In: *ICWE*. (2012) 469–472.
- [Chrissie2011] M.B. Chrissie, M. Konrad, S. Shrum, "CMMI® for Development: Guidelines for Process Integration and Product Improvement," Editorial Pearson Education. 2011.
- [Jong2008] A. Jong, A. Kolthof, "Fundamentos de ITIL, Volumen 3," Van Haren Publishing, ISBN 9087530609, 2008
- [Gutierrez] thesis
- [Escalona JSS]
- [ZRDai2004] Z. R. Dai. "Model-driven testing with uml 2.0". In *Proceedings of the Second European Workshop on Model Driven Architecture*, pages 179–187, Canterbury, UK, 2004. Computing Laboratory, University of Kent.
- [U2TP] Object Management Group (OMG) <http://utp.omg.org/>
- [Beizer1995] B. Beizer. "Black-Box Testing: Techniques for Functional Testing of Software and Systems". John Wiley & Sons, Inc., 1995.
- [BeatrizP2008] Beatriz Pérez Lamancha, Pedro Reales Mateo, Ignacio García-Rodríguez de Guzmán, Macario Polo Usaola. "Propuesta para pruebas dirigidas por modelos usando el perfil de pruebas de UML 2.0".
- [GARNET2012] Mark R. Gilbert, Karen M. Shegda, Kenneth Chin, Gavin Tay, Hanns Koehler-Kruener. "Magic Quadrant for Enterprise Content Management", 2012.
- [Stark2011] J. Stark: *Product Lifecycle Management: 21st Century Paradigm for Product Realisation*. Springer Science. DOI: 10.1007/978-0-85729-546-0_1. 2011.
- [García-García2011] García-García J.A., Cutilla C.R., Escalona M.J., Alba M., Torres J. 2011. NDT-Driver, a java tool to support QVT transformations for NDT. *The Twentieth International Conference on Information Systems Development (ISD 2011)*. ISBN: 978-1-4614-4950-8.
- [García-García2013] J.A. García-García, J. Victorio, L. García-Borgoñón, M.A. Barcelona, F.J. Domínguez-Mayo, M.J. Escalona. "A Formal Demonstration of NDT-Quality: A Tool for Measuring the Quality using NDT Methodology". *The 21st Annual Software Quality Management (SQM) conference*. 2013. Pendiente de publicación.
- [Bertolino2005] Bertolino, a., Marchetti, E., & Muccini, H. *Introducing a Reasonably Complete and Coherent Approach for Model-based Testing*. *Electronic Notes in Theoretical Computer Science*, 116, 85–97. doi:10.1016/j.entcs.2004.02.084. 2005
- [Escalona2012] M.J. Escalona, J.J. Gutierrez, M. Mejías, G. Aragón, I. Ramos, J. Torres, F.J. Domínguez, An overview on test generation from functional requirements, *Journal of Systems and Software*, Volume 84, Issue 8, August 2011, Pages 1379-1393, ISSN 0164-1212, <http://dx.doi.org/10.1016/j.jss.2011.03.051>. (<http://www.sciencedirect.com/science/article/pii/S016412121100080X>)
- [Gutierrez2012] Gutiérrez, J., Aragón, G., Mejías, M., Domínguez, F., & Cutilla, C. R. (n.d.). *Automatic Test Case Generation from Functional Requirements in NDT*. WebRE 2012.