

Aplicando Técnicas de Testing en Sistemas para la Difusión Patrimonial

J.J. Gutierrez, M.J. Escalona, M. Mejías, J.Torres

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
escalona@lsi.us.es

Resumen:

En los últimos años, la información sobre el patrimonio ha adquirido mayor interés. Turistas, investigadores y gente interesada en general demanda sistemas de información de calidad que le permitan mejorar su conocimiento sobre el legado patrimonial. Sin embargo, la amplia temática que tienen estos sistemas, así como la diversidad del público interesado, provoca que sean amplios y complejos y que sean necesarias técnicas metodológicas para su desarrollo que faciliten las tareas de validación de los resultados en las primeras fases del ciclo de vida. Este trabajo presenta la aplicación de un entorno metodológico, NDT, para el desarrollo de un sistema de información de patrimonio Mueble en Andalucía y analiza cómo la aplicación de dicho entorno permite utilizar técnicas de testing en el desarrollo de dicho sistema.

Palabras clave: Gestión patrimonial, testing, metodologías web, pruebas de sistema, pruebas funcionales.

1. Introducción

Una biblioteca digital proporciona a una comunidad de usuarios un acceso coherente a repositorios de información grandes y organizados. Las bibliotecas digitales son construidas, recogiendo y organizando la información por una comunidad de usuarios y sus funcionalidades son acordes a las necesidades de información de dicha comunidad. Las posibilidades de los usuarios para acceder, reorganizar y utilizar este repositorio están enriquecidas con las capacidades de la tecnología digital.

El desarrollo de un sistema destinado al tratamiento de bibliotecas digitales es una tarea complicada en la que se mezclan complejos requisitos de almacenamiento debido a la diversidad de formatos de información (texto, imagen, vídeo, audio, etc.) y a la existencia de varios subcampos de otros dominios como: técnicas de digitalización (OCR, OMR), integración de múltiples tipos de medios, búsqueda de información basada en el contenido (imágenes, vídeos), interacción con el usuario, manejo de información multilingüe, comercio y economía, gestión de bases de datos distribuidas.

La complejidad que pueden llegar a adquirir los sistemas para el tratamiento de bibliotecas digitales así como la importancia de que los resultados se adecuen a las necesidades del cliente, obligan a que se busquen alternativas tecnológicas y metodológicas que garanticen la calidad de los resultados.

Durante este trabajo se presenta un problema real basado en la difusión turística del patrimonio de una comunidad autónoma que ha sido desarrollado usando una metodología de desarrollo Web llamada NDT[5][6]. Este artículo se centra en estudiar cómo se puede aplicar esta metodología y cómo facilita la aplicación de una serie de técnicas de testing, ya en la fase de ingeniería de requisitos, que ayuda a encontrar errores, omisiones, inconsistencias y sobreespecificaciones en los requisitos cuando es fácil y económico corregirlas y obtener un conjunto de pruebas de sistema o funcionales que garanticen que toda la funcionalidad requerida por el sistema ha sido correctamente implementada.

Para ello, en el apartado dos se plantea el problema a resolver. En el tercer apartado se presenta una visión general de la metodología usada para el desarrollo de este sistema y el cuarto cómo se pueden usar las técnicas de testing en el mismo. Por último, se plantean las conclusiones y trabajos futuros.

2. Planteamiento del problema

La demanda creciente de la información sobre el patrimonio histórico, tanto turística como investigadora, ha movido a organizaciones y empresas dedicadas a la gestión patrimonial a buscar nuevas vías de difusión de la información que manejan.

En este sentido, el Instituto Andaluz de Patrimonio Histórico [10], ha promovido desde hace años el desarrollo de sistemas que permitan la difusión de la información vía Internet.

El Sistema de Información del Patrimonio Histórico Andaluz (SIPHA) [3] es una muestra de ello. El SIPHA se divide en dos áreas temáticas principales: el área de bienes muebles y el área de bienes inmuebles. Esta última, a su vez, se divide en el Sistema de Información de Bienes Arqueológicos, el Sistema de Información de Bienes Etnológicos y el Sistema de Información de Bienes Arquitectónicos.

Además de estos cuatro módulos principales, en el SIPHA aparecen dos elementos importantes compartidos por todas las unidades temáticas: el Sistema de Información Documental, que almacena toda la información documental, gráfica y bibliográfica referente a los bienes, y el Tesoro de Patrimonio Histórico, que almacena los términos normalizados del patrimonio.

En la figura 1 se muestra un esquema con los módulos del sistema y las interrelaciones entre ellos.

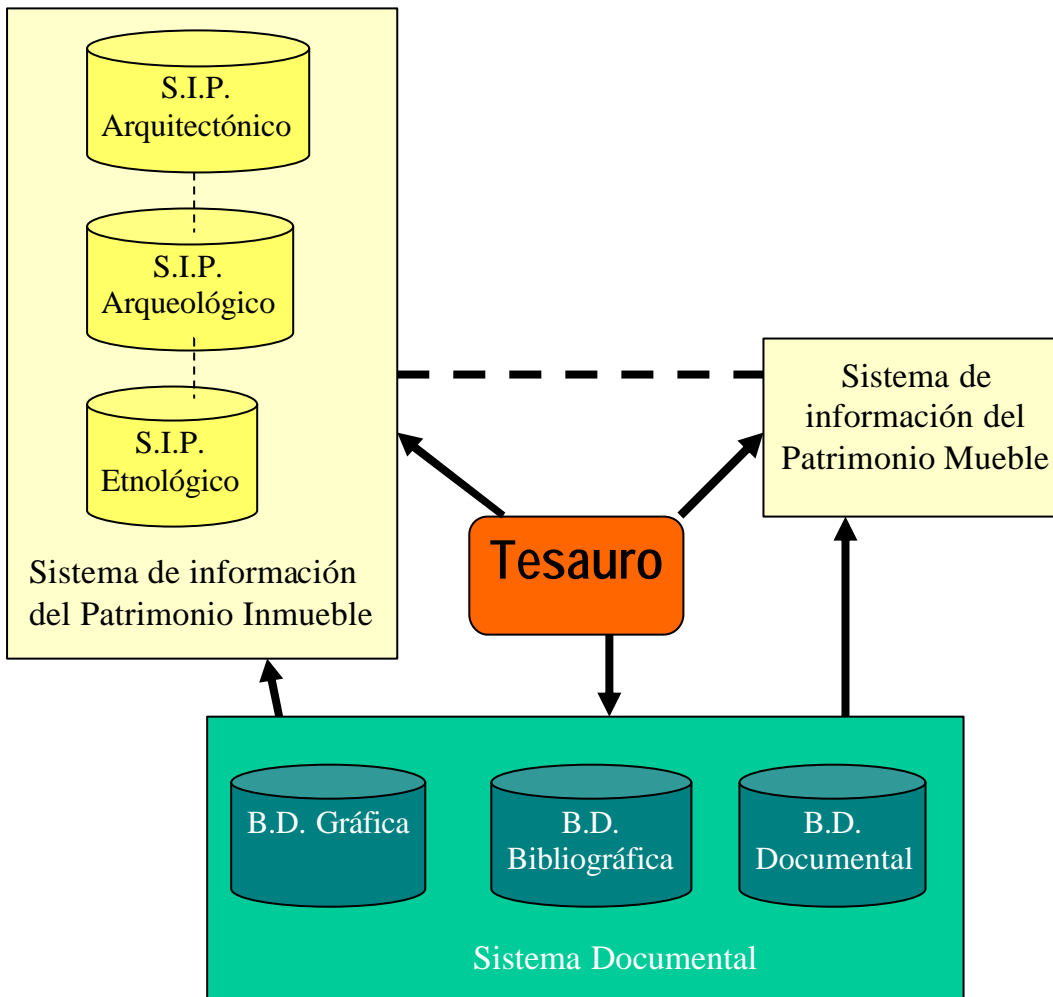


Figura 1. Estructura del SIPHA

El sistema resulta bastante complejo por la cantidad de datos, los múltiples medios con los que trabaja, las necesidades funcionales que tiene y la necesidad de que se adecue a los distintos tipos de usuario que interactúan con el sistema: arqueólogos, etnólogos, artistas, turistas, visitantes, etc.

Durante este artículo, el trabajo se centra exclusivamente en el Sistema de Información del Patrimonio Mueble[1] pues el problema completo resulta demasiado extenso para tratarlo aquí[4].

3. Visión general de NDT

NDT es un proceso metódico que se centra en las fases de requisitos y análisis y ofrece una guía sistemática para tratar en estas fases con los aspectos de navegación e interfaz abstracta. NDT entiende que en la fase de análisis se deben modelar los aspectos de navegación e interfaz abstracta. Otros aspectos como la adaptabilidad del sistema, la arquitectura o la seguridad, quedarían para la fase de diseño.

NDT no pretende ofrecer nuevos lenguajes de modelado. De los estudios comparativos realizados antes de NDT[2][7][11][13], se puede concluir en que ya existen suficientes lenguajes de modelado y modelos en análisis que han resultado válidos para modelar la navegación y la interfaz abstracta.

La aportación más importante de NDT es que ofrece una guía sistemática para el tratamiento de la navegación y la interfaz. En este sentido, se podría indicar que NDT es una propuesta orientada al proceso. NDT describe de manera detallada todos los pasos que hay que realizar para tratar los requisitos y a partir de ellos conseguir los modelos de análisis.

Tras la aplicación de las técnicas se consiguen resultados y modelos cuya nomenclatura y estructura está completamente detalladas en NDT. Además, tras la aplicación de todo el proceso, en NDT se obtienen una serie de resultados generales: el documento de requisitos del sistema, el documento de análisis del sistema y los prototipos de la interfaz abstracta. La estructura de todos ellos está descrita en NDT.

3.1 Ciclo de vida de NDT

El ciclo de vida de NDT está compuesto por dos fases: la ingeniería de requisitos y el análisis. Aunque, en principio, ambas son secuenciales, el proceso de NDT no lo es, puesto que en muchos momentos se puede realizar la vuelta atrás para corregir errores o incongruencias.

En la primera etapa de la ingeniería de requisitos se definen cuáles son los objetivos del sistema a desarrollar y en base a ellos se capturan y definen los diferentes requisitos del sistema. Los requisitos en NDT son agrupados según su carácter en requisitos de almacenamiento de información, requisitos de actores, requisitos funcionales, requisitos de interacción y requisitos no funcionales. Cada grupo de requisitos es tratado de una manera particular, adecuada a su tipología.

Una vez capturados y definidos los requisitos se pasa a la validación de los mismos. Si durante la validación se detectan errores, se vuelve a la captura y definición hasta llegar al resultado final adecuado. Este resultado final queda plasmado en el documento de requisitos del sistema.

Con el documento de requisitos, se pasa a la fase de análisis. Durante la fase de análisis se generan varios modelos. El primero de ellos es el modelo conceptual. El modelo conceptual en NDT representa la estructura estática del sistema y viene representado por un diagrama de clases. La generación de este modelo consta de dos partes, la primera de ellas es sistemática y permite conseguir un modelo conceptual básico desde los requisitos. El resultado de este proceso sistemático suele coincidir bastante con el modelo conceptual más adecuado para el sistema, pero por si se pudieran realizar mejoras que aumenten la calidad del resultado, NDT propone una segunda etapa en el proceso de creación del modelo conceptual.

En esta segunda etapa, NDT propone una serie de revisiones en las que el analista debe ir aplicando su experiencia para revisar los resultados del modelo básico. La aplicación de estas revisiones tiene dos ventajas. La primera de ellas es que, a pesar de que NDT ofrezca el proceso sistemático, también deja libertad al analista para aplicar su experiencia. Pero por otro lado, también permite detectar incongruencias y errores cometidos durante la fase de ingeniería de requisitos. Por ello, puede ser posible que durante esta actividad del análisis haya que volver a la ingeniería de requisitos a modificar los resultados.

El segundo modelo que se genera durante el análisis es el modelo de navegación. En NDT el modelo de navegación se compone de una serie de diagramas, con una notación estereotipada a partir del diagrama de clases de UML.. Los diferentes diagramas se corresponden a los sistemas de navegación para los diferentes roles de usuario que interactúan con el sistema.

El proceso de generación del modelo de navegación también consta de dos partes. La primera de ellas es sistemática y permite conseguir un modelo de navegación básico desde los requisitos. La segunda igualmente consiste en revisar el resultado del proceso sistemático para detectar incongruencias cometidas y para que el analista aplique su experiencia.. También durante esta segunda etapa se pueden detectar incongruencias en el resultado de ingeniería de requisitos que puede obligar a volver a la fase anterior para realizar revisiones.

Todos estos cambios que se pueden producir durante la generación del modelo de navegación o del modelo conceptual están controlados y detallados en NDT. NDT ofrece una guía completa de posibles modificaciones e indica cómo afectan a fases y resultados anteriores.

Cuando se tienen el modelo conceptual y de navegación, se genera en NDT la interfaz abstracta. Ésta no viene representada por un diagrama, sino por un conjunto de prototipos evaluables por el usuario.

También durante la evaluación de estos prototipos se pueden detectar errores que obliguen a volver a la etapa anterior.

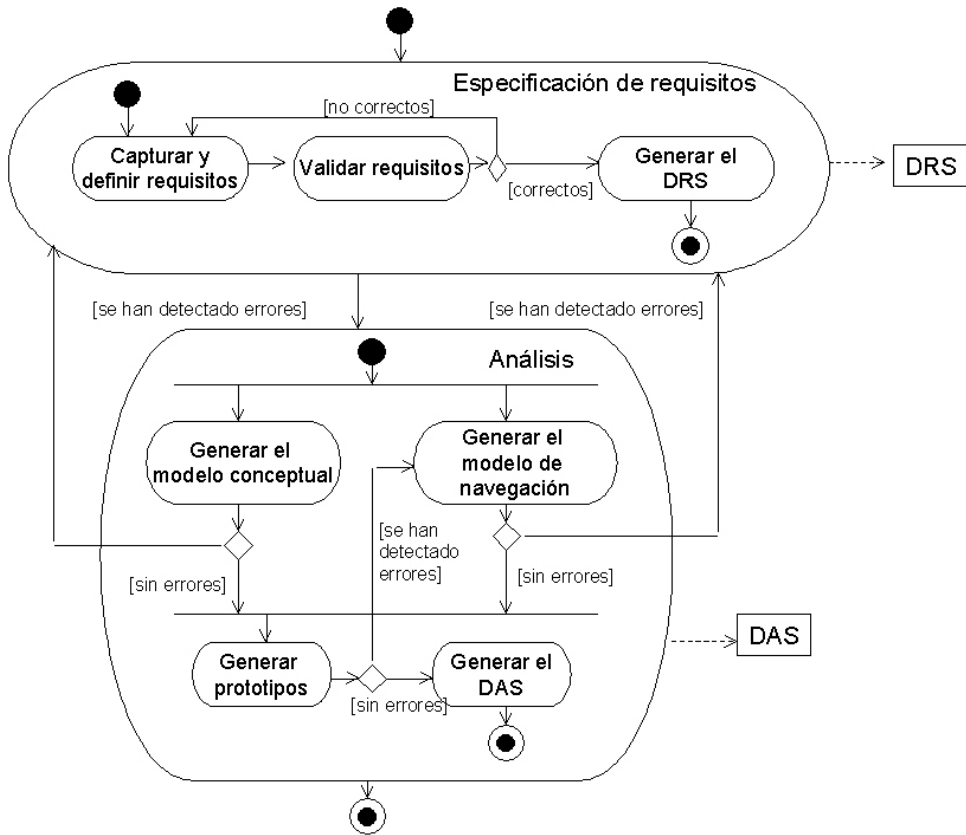


Figura 2. Descripción general de las actividades de NDT

La aplicación de NDT genera tres resultados finales:

- El documento de requisitos del sistema, donde se detallan los objetivos y requisitos que debe cumplir el sistema.
- El documento de análisis del sistema, donde se recogen el modelo conceptual y el modelo de navegación del sistema.
- Los prototipos del sistema, que muestran la estructura de la interfaz abstracta del sistema.

Estos documentos y prototipos serán la base para la realización de la etapa de diseño e implementación del sistema.

4. Técnicas de testing sobre requisitos funcionales.

4.1 Tipos de testing

Existen muchos tipos posibles de pruebas de software. Una posible clasificación de esas pruebas dependiendo de su cometido, se recoge en la tabla 1. Las fases donde se realizan cada prueba se muestran en la figura 3.

<i>Pruebas Unitarias.</i>	Prueban el diseño y el comportamiento de cada uno de los componentes una vez codificados.
<i>Pruebas de Integración.</i>	Comprueban la correcta unión de los componentes entre sí a través de sus interfaces, y si cumplen con la funcionalidad establecida
<i>Pruebas del Sistema.</i>	Prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción.
<i>Pruebas de Implantación.</i>	Comprueba el correcto funcionamiento del sistema dentro del entorno real de producción.
<i>Pruebas de Aceptación.</i>	Verifican que el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema den el visto bueno definitivo.
<i>Pruebas de Regresión.</i>	El objetivo es comprobar que los cambios sobre un componente, no generan errores adicionales en otros componentes no modificados.

Tabla 1. Clasificación de las pruebas del software

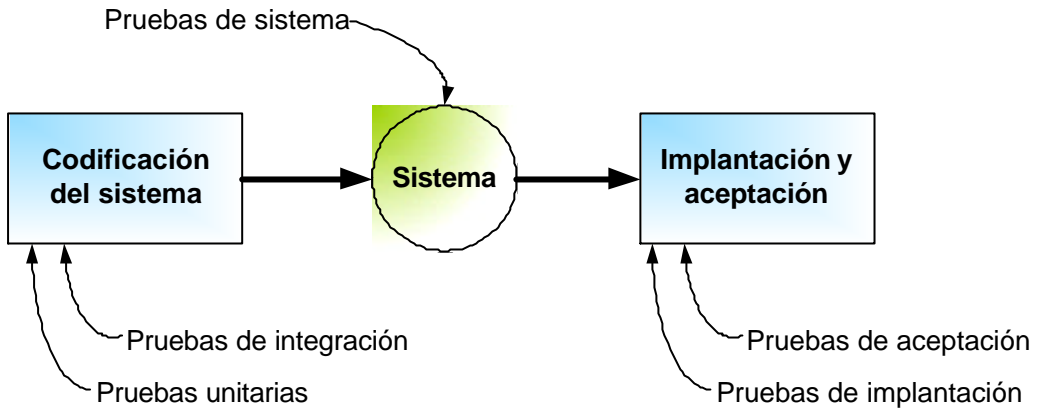


Figura 3. Fases donde se realizan los distintos tipos de pruebas

Las pruebas unitarias y de integración no pueden realizarse hasta que no se dispone de componentes ya codificados, al igual que las pruebas de implantación y aceptación no pueden realizarse hasta que se tiene el sistema completo y se instala en su entorno de producción. Pero la planificación de las pruebas del sistema puede comenzar antes de que el sistema esté completo.. Como el objetivo de las pruebas de sistema es comprobar que todo lo que se está desarrollando cumple con la funcionalidad recogida en los casos de uso o escenarios, la planificación de estas pruebas y el diseño de los casos de prueba pueden comenzar tan pronto como estén disponibles las especificaciones funcionales. La planificación y diseño de pruebas en las primeras fases de desarrollo, además, permite encontrar errores, omisiones, inconsistencias y sobreespecificaciones en los requisitos funcionales cuando aún es fácil y barato corregirlas.

Para que el proceso de prueba del sistema sea eficaz debe estar integrado dentro del propio proceso de desarrollo y, como cualquier otra fase del desarrollo, debe realizarse de manera sistemática, por lo que el factor experiencia o intuición debe minimizarse. Esto se consigue a través de metodologías que guíen el proceso de desarrollo de pruebas del sistema a partir de requisitos funcionales.

4.2 Propuestas existentes para la obtención de casos de prueba para pruebas del software

Actualmente existen varias propuestas para obtener pruebas del sistema a partir de los requisitos en las fases tempranas del desarrollo de software,

Propuesta.	Descripción.
<i>SCENT [12]</i>	Ofrece un método muy completo para manipular y organizar los requisitos funcionales en escenarios de uso y derivar casos de prueba del sistema a partir de ellos.
<i>AGEDIS [8].</i>	Abarca tanto la elaboración de un prototipo del sistema como la generación y ejecución de las pruebas. Cuenta, además, con un conjunto de herramientas para automatizar el proceso.
<i>Generating Test Cases From use Cases [9].</i>	Desarrolla un método basado en tres pasos para obtener un conjunto de casos de prueba del sistema a partir de casos de uso descritos en lenguaje natural.
<i>UML-Based Statistical Test Case Generation [14]</i>	Está orientada a obtener un conjunto de casos de prueba para realizar pruebas de carga en función de las probabilidades de cada caminos de ejecución para cada requisito funcional.
<i>Use Case Path Anmalisys [15].</i>	Analiza todos los caminos de ejecución para un caso de prueba, los puntúa y selecciona el conjunto mínimo que deben traducirse a pruebas del sistema.

Tabla 2. Propuestas para la obtención de casos de prueba a partir de requisitos funcionales

Aunque cada propuesta hace especial hincapié en algunos aspectos concretos, todas tienen unos principios comunes que enumeramos a continuación:

1. El objetivo de estas propuestas es obtener un conjunto completo de pruebas del sistema que permitan garantizar que el sistema software cumple con la especificación funcional dada, lo cual permite asegurar su calidad.
2. Todas parten de los requisitos funcionales del sistema y todas permiten comenzar a desarrollar los casos de prueba del sistema en cuanto se dispongan los requisitos funcionales.
3. Todos usan el análisis de los caminos posibles, bien mediante la descripción textual de los pasos del escenario o caso de uso o mediante diagramas de estado.

4. Los requisitos funcionales no tienen que cumplir de principio ningún requisito formal. A partir de una breve descripción en lenguaje natural ya se puede comenzar a trabajar.
5. La derivación de pruebas del sistema a partir de los requisitos funcionales se realiza de manera automática y sistemática. Todas las propuestas son susceptibles de automatización mediante herramientas software.
6. La aplicación de estas propuestas a los requisitos funcionales ayuda a validarlos, comprobando si son correctos y completos en las primeras fases de desarrollo.

4.3 Ventajas e inconvenientes de aplicar las propuestas de generación de casos de prueba al sistema de bienes muebles.

Uno de los aspectos no contemplados en un elevado número de propuestas metodológicas es el que tiene relación con la definición de las pruebas que se han de realizar a un sistema para garantizar que el mismo responde a la definición de requisitos que para él se hizo. Aunque la definición concreta de las pruebas que se deben aplicar a un sistema deberá concretarse en el momento de concretar su diseño, toda propuesta metodológica debe definir las posibles estrategias a seguir, las técnicas a utilizar para realizarlas y el momento de hacer uso de cada una de ellas. En el marco de la propuesta para el desarrollo de aplicaciones basadas en bibliotecas digitales se debe elaborar, diseñar e implementar un plan de pruebas. Este plan de pruebas no solo debe recoger las pruebas a realizar, además debe indicar el orden de realización. Una vez realizado, hay que ejecutar el plan de pruebas y elaborar una memoria de resultados del mismo.

Las ventajas principales son que garantiza la calidad del sistema y permite validar requisitos en fase temprana, especialmente en este caso donde podemos encontrar un número elevado de casos de uso a medida que vamos detallando la funcionalidad del sistema.

El principal inconveniente es que estas propuestas aún han sido poco estudiadas y no hay una propuesta uniforme.

4.4 Aplicación de las propuestas de generación de casos de prueba al sistema de bienes muebles.

En el sistema de gestión del patrimonio mueble existe un gran número de requisitos funcionales; un conjunto representativo de estos requisitos se muestra a continuación:

- RF-01: La aplicación debe permitir dar de alta nuevos bienes, eligiendo la provincia, el municipio y el inmueble, se asignará automáticamente el número de mueble de forma secuencial.
- RF-02: La aplicación debe permitir seleccionar un bien mueble y modificar sus datos
- RF-03: La aplicación debe permitir seleccionar un bien mueble y modificar sólo sus datos museográficos
- RF-04: La aplicación debe permitir realizar consultas sobre los datos mediante los campos de identificación y descripción.

No es necesario disponer de todos los requisitos funcionales para poder aplicar cualquiera de las propuestas, sino que pueden comenzar a derivarse casos de prueba en cuanto se comiencen a tener requisitos validados, al mismo tiempo que se continúan identificando y definiendo más requisitos.

Una vez identificado un requisito funcional es necesario detallarlo antes de poder aplicar cualquiera de las técnicas de testing comentadas en el punto 4.1. Para cada requisito funcional se ha de describir, al menos, las precondiciones y poscondiciones, los actores implicados y sus roles, los pasos de la secuencia normal de ejecución, las excepciones y las dependencias con otros requisitos funcionales si existieran.

Con esta información ya es posible comenzar a aplicar cualquiera de las propuestas vistas. En el ejemplo concreto del sistema de bienes muebles, dada la alta complejidad y gran tamaño del sistema, la metodología más adecuada sería AGEDIS [8] ya que está desarrollada para sistemas grande y complejos y ofrece un conjunto de herramientas que automatizan las tareas. Sin embargo, esta elección tiene las desventajas de necesitar una inversión inicial para la adquisición de herramientas y formación de los ingenieros de pruebas. Además esta metodología ha sido desarrollada para sistemas de control por encima de sistemas de almacenamiento y tratamiento de información.

También, como punto de partida, se pueden optar por comenzar con alguna de las propuestas más simples, como [9] y [15], las cuales pueden aplicarse con una preparación mínima por parte de los ingenieros de prueba. Para un sistema de alta complejidad como este, a la larga esta no es una solución adecuada, pero sí un buen primer paso para adquirir experiencia e ir evolucionando hacia propuestas más complejas.

Cualquiera de las propuestas estudiadas se acopla fácilmente al ciclo de vida NDT. Concretamente, en la fase de Especificación de Requisitos de NDT se identifican y validan los requisitos funcionales, por lo que, en paralelo, se puede comenzar el proceso de obtención de pruebas del sistema en cuanto se obtengan requisitos validados. Este proceso también permite realizar una validación adicional de los requisitos.

5. Conclusiones

Este trabajo ha presentado un problema real, el Sistema de Información del Patrimonio Histórico Andaluz, y la visión general de la metodología, NDT, que se ha aplicado para su desarrollo. Tras esto ha ofrecido una visión general de las técnicas de testing que se pueden aplicar al tener el sistema desarrollo de manera normalizada mediante NDT y ha analizado las ventajas de la aplicación de las técnicas.

En la actualidad, NDT ha sido aplicado a varios desarrollos dentro del Instituto Andaluz de Patrimonio Histórico. La base que estos desarrollos ofrecen permite que en la actualidad, tengamos planteado como trabajo futuro la evaluación de las diferentes técnicas de testing en estos entornos, para seleccionar las más adecuadas y permitir que sean aplicadas desde el principio en el desarrollo de nuevos sistemas.

Bibliografía

- ARENILLAS, J.A., MUÑOZ, V., ESCALONA, M.J. *La base de datos Bienes Muebles...Arqueológico*. Capítulo V. Pág.71-97. ARQUEOS. Sistema de Información del Patrimonio Arqueológico de Andalucía. Cuadernos Técnicos. N°9. Consejería de Cultura. Junta de Andalucía. Sevilla, España. 2002
- BARRY, C., LANG, M *A Survey of Multimedia and Web Development Techniques and Methodology Usage*. IEEE Multimedia. 52-56. April-June 2001.
- CORDERO, J.M., ESCALONA, M.J., TORRES, I, MEJÍAS, M., GASCA, R.M. *Aplicación de los sistemas de tratamiento de bibliotecas digitales a la gestión de Patrimonio Histórico*. Número monográfico. Estudios Turísticos N° 146. pp. 37-47. Instituto de Estudios Turísticos. Madrid, España. Diciembre 2000.
- ESCALONA, M.J., TORRES, J., MEJÍAS, M. *Aplicación de los sistemas de tratamiento de bibliotecas digitales al Sistema de Información del Patrimonio Histórico Andaluz*. Boletín Trimestral del Instituto Andaluz de Patrimonio Histórico, N° 32. pp. 205-209. Sevilla, España. Septiembre 2000.
- ESCALONA M.J, MEJÍAS M, TORRES J, REINA A.M. *The NDT Development Process*. Proceedings of IV International Conferences on Web Engineering. ICWE 2003. LNCS 2722. pp. 463-467. Springer Verlag 2003
- ESCALONA, M.J., MEJÍAS, M., TORRES, J., REINA, A.M. *NDT: una técnica para el desarrollo de la navegación*. 5º WorkShop Iberoamericano de Ingeniería de Requisitos y Ambientes Software. pp. 305-315. La Habana. 2002
- ESCALONA, M.J., KOCH, N. *Requirements Engineering for Web Applications: A Comparative Study*. Journal on Web Engineering, Vol.2 N°3, pp. 193-212. 2004. Rinton Press
- HARTMAN, A. 2004 AGEDIS Final Project Report AGEDIS Consortium Internal Report.
<http://www.agedis.de/>
- HEUMANN, J. 2002. *Generating Test Cases from Use Cases*. Journal of Software Testing Professionals.
- INSTITUTO ANDALUZ DE PATRIMONIO HISTÓRICO. D.G. de Bienes Culturales. Junta de Andalucía.
<http://www.juntadeandalucia.es/cultura/iaph/>
- KOCH, N. *Software Engineering for Adaptive Hypermedia Applications*. Ph. Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck Publishing Company, Munich. Germany. 2001.
- RYSER, J., GLINZ, M. 2003. *Scent: A Method Employing Scenarios to Systematically Derive Test Cases for System Test*. Technical Report 2000/03, Institut für Informatik, Universität Zürich.

RETSCHITZEGGER, W. & SCHWINGER, W. *Towards Modeling of Data Web Applications - A Requirements Perspective* Proceedings of the American Conference on Informating Systems AMCIS 2000, Vol 1, pp. 149-155.

2000

RIEBISCH, M., PHILIPPOW, I., ILMENAU, M.G. 2002. *UML-Based Statistical Test Case Generation*. Technical University, Ilmenau, Germany

AHLOWALIA, NARESH. 2002. *Testing From Use Cases Using Path Analysis Technique*. International Conference On Software Testing Analysis & Review