
Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental

User-oriented acceptance testing: an agile context for document management projects

**José Ponce GONZÁLEZ , Francisco José DOMINGUEZ MAYO,
Javier Jesús GUTIÉRREZ RODRÍGUEZ, María José ESCALONA CUARESMA**

Grupo de Investigación de Ingeniería Web y Testing Temprano (IWT2), Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Avda. Reina Mercedes s/n. 41012 Sevilla, {josepg, fjdominguez, javierj, mjescalona}@us.es

Resumen

Las pruebas de aceptación representan aquella fase del ciclo de vida de desarrollo de software en el que el equipo de desarrollo y el área usuaria de un sistema de información tienen que garantizar que el sistema desarrollado se corresponde con los requerimientos definidos. En equipos multidisciplinares, como en el caso de los sistemas de gestión documental, que involucra a documentalistas, archiveros, ingenieros informáticos, etc., es una fase que hay que planificar de manera adecuada y a la que hay que dotar con los mecanismos de gestión adecuados para garantizar su correcta ejecución. En este trabajo se propone la adaptación de un mecanismo ágil, basado en la metodología Scrum, que da una importancia alta a esta fase, de manera que se propone ir abordándola de manera incremental, no dejando la ejecución de dicha fase para el final del desarrollo, ofreciendo así mecanismos de detección precoz de inconsistencias funcionales. Esta propuesta ha sido desarrollada en el contexto del proyecto THOT (Proyecto de innovación de la gestión documental aplicada a expedientes de contratación de servicios y obras de infraestructuras de transporte), desarrollado por la Universidad de Sevilla para la Agencia de Obra Pública de la Junta de Andalucía.

Palabras clave: Gestión documental. Sistema web. Pruebas de aceptación. Diseño centrado en el usuario. Metodología ágil. Agencia de Obra Pública de la Junta de Andalucía. Expedientes de contratación de servicios y obras de infraestructuras de transporte.

1. Introducción

Dentro del contexto de las pruebas (Ponce, 2010a; Ponce, 2010b), una de las más importantes son las de aceptación (*user's tests*). Son aquellas pruebas que son diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados en la fase de análisis para cubrir todo ese espectro, y ejecutadas por el propio usuario final, no por todos evidentemente, pero sí por una cantidad de usuarios finales significativa que de validez y conformi-

Abstract

Acceptance tests represent the phase of the software development life cycle in which the development team and the user area of an information system must ensure that the developed system fulfils the defined requirements. In multidisciplinary teams, like is the case of document management systems that involve filmmakers, archivists and computer engineers among others, this is a phase that must be properly planned and must provide the most suitable management mechanisms to ensure proper execution. This paper presents the adaptation of an agile mechanism based on the Scrum methodology, giving high importance to this phase. So, it is proposed to perform this phase incrementally, preventing this phase to be executed only at the end of development and providing mechanisms for early detection of functional inconsistencies. This approach has been developed within the context of THOT (Innovation project management applied to document records of recruitment services and transport infrastructure projects), developed by the University of Seville for the Public Works Agency project of the Junta de Andalucía.

Keywords: Document management. Web system. Acceptance testing. User centered design. Agile methodology. Public Works Agency of the Andalusian Government. Recruitment services and transport infrastructure projects files.

dad al producto que se les está entregado en base a lo que se acordó inicialmente.

Dependiendo de la complejidad del sistema a probar, si está o no dividido por módulos, etc., la realización de dichas pruebas se ejecuta de forma diferente; aunque de manera general las pruebas de aceptación pueden afrontarse mediante dos tipos de procedimiento para realizarlas: pruebas alfa y pruebas beta.

La prueba alfa es aquella en la que se le entrega a un usuario final todo el producto terminado,

junto a su documentación correspondiente para que éste, en presencia del desarrollador y en entornos previamente preparados para el proceso de dichas pruebas, vaya informando de las inconsistencias y errores que detecte.

Por otro lado, la prueba beta es la que se le proporciona a usuarios finales, situados en lugares concretos de los puestos de trabajo donde finalmente será el software implantado, para que sea otra vez el usuario y sin contar con la presencia del equipo de desarrollo, el que de nuevo vuelva a emitir unos informes de resultados e impresiones de la aplicación o sistema software desarrollado.

Independiente del proceso seguido, es fundamental que la calidad tanto del código como de la documentación aportada al usuario sea alta y se corresponda con los parámetros adecuados para el software que concretamente se esté desarrollando.

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de las pruebas de aceptación en el desarrollo de sistemas de información. Las más tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Aunque también existen tendencias que se focalizan en otras dimensiones, como por ejemplo el factor humano o el producto software, aunque siempre sin dejar de lado la documentación. Es en esta última donde se centra la propuesta de metodología que se presenta en este trabajo.

Esta propuesta ha sido desarrollada en el contexto del Proyecto de innovación de la gestión documental aplicada a expedientes de contratación de servicios y obras de infraestructuras de transporte (THOT) (Proyecto THOT, 2014). Este proyecto es un proyecto de investigación, desarrollo e innovación que explora los mecanismos para la gestión de procesos en el contexto de la administración pública haciendo uso de la e-administración y basándose en estándares de referencia como el esquema ENI (Esquema Nacional de interoperabilidad) (ENI, 2014).

La propuesta presentada ofrece un marco de referencia novedoso y que aporta grandes ventajas en el contexto de proyectos como THOT. Es una propuesta que permite:

1. Involucrar al usuario en las pruebas de aceptación en fases tempranas, lo que permite una detección precoz de los errores funcionales, reduciendo así su impacto a la hora de resolverlos.

2. Facilitar a equipos multidisciplinares y a usuarios no expertos su involucración en el desarrollo, aportando su conocimiento y experiencia desde el inicio y de una forma natural.

Para ello, se propone una adaptación de la metodología Scrum (Scrum, 2014) y se focaliza en el uso de la Ingeniería Guiada por Modelos (Escalona y Aragón, 2008) como solución técnica para la generación de pruebas tempranas.

El artículo se organiza de la siguiente manera. En la sección 2 se amplía la información relacionada con las metodologías de desarrollo ágiles y los estándares actuales sobre pruebas de software. En la sección 3, se contextualiza el trabajo dentro del marco del proyecto THOT y se explica su alcance. La sección 4 detalla la metodología ágil Scrum, y su adaptación al proyecto THOT. En la sección 5, se detalla la organización y estructura de los procesos de pruebas de aceptación seguidos en el contexto de la metodología anterior. Y, finalmente, se explican las conclusiones más relevantes del artículo.

2. Metodologías y estándares de pruebas

En esta sección se presentan los contextos tecnológicos de referencia de nuestra propuesta. Inicialmente se presenta una visión general de las metodologías ágiles y, a continuación, se ofrece una revisión sobre los estándares de prueba de referencia.

2.1. Metodologías ágiles

Tras una reunión celebrada en Utah (EEUU) en febrero de 2001 (Canós, 2003; Agile Alliance, 2014), nació el término ágil aplicado al desarrollo de software. En esta reunión participaron un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que pudieran surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó "The Agile Alliance", una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones a adoptar dichos conceptos. El punto de partida fue el *Manifiesto para el desarrollo de*

software ágil (Manifiesto for Agile Software Development, 2014). Este manifiesto se enfoca:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Las personas son el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- A desarrollar software que funcione de manera adecuada y orientado a los requisitos, más que a conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- A la colaboración con el cliente más que a la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- A responder a los cambios más que a seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Estos valores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto a metas a seguir y organización del mismo. Los principios son:

1. Satisfacer al cliente mediante entregas tempranas y continuas de software que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.

4. Los responsables de negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. A intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

2.2. Estándares sobre pruebas

Como ya se ha comentado previamente, en la fase de validación se intenta que el software sea entregado con la mayor calidad posible, entendiendo como producto de calidad aquel que cumpla con las características marcadas por algunos estándares, entre los que destaca el estándar ISO/IEC 25000 SQuaRE. El objetivo general del estándar ISO/IEC 25000 SQuaRE es organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requerimientos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software. La serie ISO/IEC 25000:2005 reemplaza a dos estándares relacionados: ISO/IEC 9126 e ISO/IEC 14598. Por tanto, el objetivo de la fase de pruebas es que el producto final acabe cumpliendo las expectativas creadas por el mismo, ciñéndose de esta manera al estándar establecido.

En cuanto a estándares de pruebas se refiere, existen algunos —como BS 7925 o IEEE Std 1008 (IEEE Standard 1008-1987, 2014)— primordialmente enfocados a pruebas unitarias que son llevadas a cabo en fases tempranas dentro de la fase de pruebas. Sin embargo, existen áreas no cubiertas por los estándares

anteriores como son aspectos organizativos, proceso y gestión de las pruebas, técnicas funcionales y no funcionales, etc. Para ello, se ha desarrollado un nuevo estándar, ISO/IEC/IEEE 29119 (ISO/IEC/IEEE 29119-1:2013, 2014). Su objetivo principal es el de unificar los anteriores estándares en uno solo para intentar cubrir la mayoría de áreas no cubiertas por los mismos hasta el momento. En nuestro contexto, nos basaremos en los principios de este estándar.

3. El proyecto THOT

El proyecto THOT pretende dar un salto cualitativo abarcando diferentes disciplinas de investigación e innovación como son principalmente: políticas de gestión documental, políticas de administración electrónica, políticas de difusión e integración en entornos Web (aspectos todos ellos tratados e investigados de manera profusa por diferentes grupos y foros de investigación, tanto nacionales como internacionales) que permiten a las organizaciones disponer de un marco de referencia común para la gestión documental, así como cubrir la necesidad que tienen de disponer de una gestión integral que complementa los procesos de negocio de principio a fin.

Para ejemplificar la solución propuesta, el proyecto ofrece un prototipo que permite gestionar los expedientes de contratación de servicios y obras de la propia Agencia.

En este sentido, ofrece unos beneficios básicos que pueden centrarse en tres entes principalmente:

1. A nivel de la propia Agencia de Obra Pública de la Junta de Andalucía, el proyecto pretende ofrecer una solución de administración electrónica orientada a entender los expedientes en general, aunque el proyecto esté particularizado sobre los expedientes de contratación de servicios y obras de infraestructura de transporte, tanto desde el punto de vista de gestión, como desde el punto de vista de la gestión documental. Todo ello, enmarcado dentro de las políticas tanto autonómicas de la Junta de Andalucía como nacionales e internacionales.
2. A nivel de la Administración Pública Andaluza, el proyecto pretende estar totalmente integrado dentro del marco andaluz para la gestión pública y seguir las políticas administrativas. Los resultados, tal y como se muestran en este plan de difusión, quedarán a disposición de la administración, pudiendo ser una fuente de conocimiento y de expe-

riencias para otros sectores de la Administración local.

3. A nivel social. El beneficio que puede otorgar el proyecto THOT a nivel social es uno de los más relevantes del proyecto. La agilización de los procesos documentales así como de la tramitación de expedientes, repercute de manera directa sobre la sociedad puesto que esto deriva en un aumento de la eficacia de la Administración, y, por ende, del servicio que recibe la ciudadanía.

Pero además, el proyecto THOT, al ser un proyecto de investigación está orientado a aportar una serie de resultados de interés a la comunidad científica que se medirá por las publicaciones y actividades de difusión en el marco de la investigación que se pretenden contemplar.

En este sentido, este proyecto realiza una mejora importante en los siguientes campos:

- Prestaciones, ya que los resultados del proyecto mejoran las prestaciones en el campo de la gestión documental y de expedientes así como en su integración con las políticas de e-administración y los marcos normativos de referencia. Repercutiendo, directamente, en una mejora para la sociedad.
- Calidad, puesto que los resultados estarán sistematizados y automatizados en su gran mayoría, se garantiza la calidad tanto del proceso como del servicio.
- Capacidad tecnológica, que se verá beneficiada con los resultados del proyecto haciendo más competitivo de manera directa el servicio de la propia Agencia de Obra Pública de la Junta de Andalucía, como a los beneficiarios del mismo.
- Costes, pues los aspectos anteriores repercuten en un beneficio de los costes de ejecución que conllevan los procesos en él contemplados.
- Impacto laboral y ambiental, que vendrá reflejado por la mejora en la productividad de las personas que utilicen los resultados del proyecto. Cabe destacar que, en la búsqueda de una solución capaz de soportar el expediente digital, se pretende ofrecer una mejora muy relevante a nivel ambiental.

A modo de ejemplo, y teniendo en cuenta que no es el objetivo de este trabajo, en la figura 1 se presenta un ejemplo de pantalla del prototipo desarrollado dentro de THOT. En esta pantalla se muestra el cuadro de clasificación soportado por el mismo y se ejemplifica la interfaz con la trabaja el usuario en esta línea. Como puede

verse, es un proyecto orientado al usuario con una interfaz web que soporta las políticas de calidad y accesibilidad definidos en el contexto de la Agencia. Dicho cuadro ofrece las posibilidades de acceder a la tramitación de expedientes así como a la gestión organizada de documentos para cada una de las fases por las que se encontrará un determinado expediente.

Para el desarrollo del piloto, se cuenta con la colaboración de la empresa Tecnomcom, que es la que aporta el equipo de desarrollo. Como único usuario final del proyecto tendríamos al funcionariado de la propia Agencia.

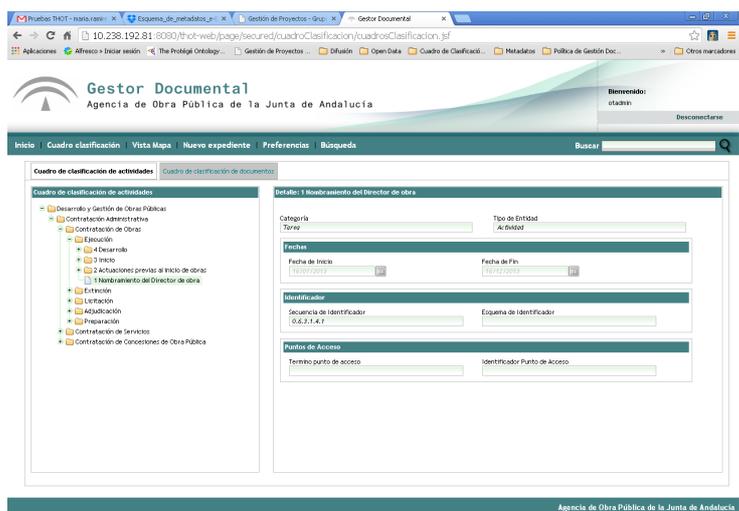


Figura 1. Ejemplo de pantalla con el cuadro de clasificación de actividades del sistema

4. Metodología para el desarrollo de la solución tecnológica y las pruebas de aceptación

A lo largo de este apartado, se muestra cómo se ha llegado a la solución metodológica que está siendo aplicada en el marco del proyecto THOT.

4.1. Contexto de partida: NDT & Scrum

Antes de entrar en detalle en la fase de pruebas, hay que comentar que el proyecto se inició haciendo uso de una metodología de desarrollo definida por el propio grupo de investigación: La metodología NDT (Navigational Development Techniques) (Grupo IWT2, 2014). NDT establece rigurosamente las actividades involucradas, los artefactos a producir, y las herramientas y notaciones que se deben usar.

NDT es una propuesta metodológica que se mueve dentro del entorno de la ingeniería del software orientada a la Web (Escalona y Aragón, 2008) y que centra en los principios de la

Ingeniería guiada por modelos para el desarrollo de software.

Además, ofrece un conjunto de herramientas denominado NDT-Suite (Grupo IWT2, 2014) y que ofrece un soporte integral al desarrollo. En el proyecto THOT NDT, dicho conjunto ha sido aplicado de manera tradicional en las fases de requisitos y análisis. Sin embargo, es en la fase de diseño y construcción donde se ha adaptado hacia una propuesta centrada en el factor humano.

Para ello, se ha seguido la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

En este sentido, nuestro referente ha sido la metodología de desarrollo ágil Scrum (Scrum, 2014). Esta metodología define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos o dicho de otra manera, para aquellos proyectos que son susceptibles de cambios. Sus principales características se pueden resumir en dos:

- El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 15-20 días laborables. El resultado de cada *sprint* es un incremento de funcionalidad que se muestra al cliente y le aporta valor.
- Las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

4.2. Las pruebas tempranas de aceptación

La idea general que se tiene para validar aplicaciones o sistemas informáticos es realizar un conjunto de pruebas de diferente índole y tipo, desarrolladas por un conjunto de personas pertenecientes a las diversas entidades implicadas, ejecutadas por personas que de nuevo pueden pertenecer o no a esas mismas o diferentes entidades, que por uno u otro lado componen toda esa maraña que rodea al complejo proceso de creación de software, en el cual una de las fases más importantes del ciclo de vida por donde va pasando dicho software es el estudio, diseño, ejecución y validación de estas pruebas, que van a dar una visión o valoración muy im-

portantes al software que finalmente se está construyendo.

Las pruebas de aceptación, dentro de este contexto, son aquellas en las que los usuarios o expertos en el negocio validan los requisitos funcionales. Una propuesta temprana de dichas pruebas se refiere al hecho de que las pruebas se definen directamente desde los requisitos. Esto implica que el equipo de desarrollo, una vez definidos los requerimientos y validados éstos con el usuario, no son los encargados de definir o diseñar dichas pruebas, sino que estas se generan de manera sistemática, incluso automática si es posible, desde los requisitos del usuario.

En nuestra propuesta, dichas pruebas son generadas de manera automática, haciendo uso de la herramienta NDT-Suite y siguiendo los mecanismos definidos en Gutiérrez (2008) y Escalona (2011).

5. Organización y estructura de los procesos de pruebas de aceptación

Las pruebas de aceptación que se abordan en THOT pretenden cubrir las dimensiones de calidad mostradas a continuación basándonos en Piattini (2004) y Pressman (2002).

El *contenido* se evalúa tanto a nivel sintáctico como semántico. En el primero se harán revisiones de vocabulario, puntuación, gramática, etc. y en el segundo se valora la corrección de la información presentada, la consistencia, así como la falta de ambigüedad en lo aportado.

La *función* se probará para descubrir errores que indiquen la falta de conformidad por parte del cliente. Cada función de la aplicación se valorará en su corrección, inestabilidad así como la conformidad con diferentes estándares de programación.

Respecto a la *estructura*, se valorará que garantice adecuadamente el contenido y la función de la aplicación, que sea extensible y que pueda soportarse conforme se agregue nuevo contenido o funcionalidad.

La *usabilidad*, para asegurarse que la aplicación es capaz de soportar a cada tipo de usuario que se pueda conectar a ella, pudiendo aprender y aplicar toda la sintaxis y semántica de la navegación requerida.

La *navegabilidad* se probará para que toda la estructura de navegación esté correcta y descubrir errores tan simples como usuales de enlaces muertos, inadecuados o erróneos.

El rendimiento, la compatibilidad, seguridad son otros aspectos a tener en cuenta en dichas revisiones para que la satisfacción del cliente sea máxima y la confianza en el producto la más alta posible.

El proceso de gestión de las pruebas dentro del proyecto se define de la siguiente manera:

Paso 1. Definiendo el sprint

El equipo de trabajo —que involucra tanto al equipo de desarrollo de TecnoCom, al área usuaria y a los investigadores del grupo— define las historias de usuario que se van a abordar en cada sprint.

Como se ha comentado en la sección 4.1 para la definición de requisitos se usó la metodología NDT. Ésta propone definir las necesidades funcionales del sistema en base a casos de uso, por lo que en la definición del sprint lo que realmente se hace es determinar cuáles son los casos de uso que se van a diseñar e implementar durante el mismo.

En el proyecto THOT se ha definido una duración del sprint de tres semanas y se han desarrollado un total de nueve sprints de manera inicial, aunque, como la propia metodología Scrum propone, se van revisando con la evolución del sistema.

Paso 2. Desarrollando el sprint

El equipo de desarrollo, compuesto por los técnicos de TecnoCom, desarrolla durante tres semanas las historias de usuario (casos de uso) previstos en el Sprint. En este contexto, se desarrolla el código, se documenta y se realizan las pruebas unitarias y de integración.

Pasadas las tres semanas, se realiza una reunión de seguimiento de presentación y aceptación del sprint. En esta reunión se analizan los casos de uso cubiertos en el sprint y se analizan los aspectos relativos a la funcionalidad cubierta. En esta reunión, se presentan además los resultados de las pruebas unitarias y de integración. Usando NDT-Suite, se generan los casos de prueba de aceptación de los casos de uso que han sido cubiertos en el Sprint. Esto lo realiza el grupo de investigadores de la Universidad de manera independiente al código, tal y como se propone en los estándares de referencia como la ISO/IEC/IEEE 29119.

Paso 3. Realización de pruebas de aceptación

Una vez realizada la presentación del sprint, TecnoCom implanta el nuevo paquete de funcionales en el entorno de preproducción instalado en el destino final y se comienzan las pruebas de aceptación. En éstas, se siguen los casos de

prueba generados con NDT-Suite, de una manera independiente al desarrollo y sin participación por parte de TecnoCom. En dichas pruebas se ven implicadas, recursos del grupo de investigación que son principalmente los encargados de ir realizando las pruebas tal y como se indica en los diferentes planes de pruebas y apoyados en todo momento como soporte, con una representación del área usuaria que, ante dudas de resultados de las pruebas, podría dar luz a la decisión final sobre la posible incidencia a notificar.

Si se detectan errores o incongruencias, se registran en una herramienta de gestión de incidencia para ser considerados dentro del siguiente sprint o en el que sea oportuno. La herramienta de gestión de incidencias utilizada,

una extensión de Redmine, permite hacer un seguimiento efectivo de la resolución de incidencias.

A modo de ejemplo, en la siguiente tabla se muestran las incidencias detectadas en las baterías de pruebas correspondientes a los cuatro primeros sprint.

En el gráfico de la izquierda se muestran las historias de usuario detectadas con el índice de pruebas que han provocado incidencias (KO), de las que han resultado adecuadas (OK) y de las que, por alguna razón no se han podido realizar (por ejemplo, por no estar dado de alta el usuario de prueba). En el gráfico de la derecha se muestran los mismos resultados de una manera más visual.

Etiquetas de fila	KO	OK	Pdte.	Total general
HU01	3	1	1	5
Gestión dinámica de listados	2			2
Login de usuario	1	1		2
Navegación por cuadro de clasificación documentos			1	1
HU02	4	2		6
Detalle de expediente tabular	1			1
Edición de expediente	3	1		4
Navegación por cuadro de clasificación actividades		1		1
HU03	3	2		5
Búsqueda de expedientes	3			3
Detalle de expediente árbol		1		1
Preferencias de usuarios		1		1
HU04	4		1	5
Creación de expedientes	3			3
Detalle de expediente Timeline			1	1
Gestión roles usuario (seguridad basada en roles)	1			1
HU05			1	1
Vista de expedientes en mapa			1	1
Total general	14	5	3	22



Figura 2. Ejemplo de informe de seguimiento de pruebas de aceptación

6. Conclusiones

El proceso de desarrollo de software en proyectos multidisciplinares requiere que haya una participación efectiva de todos los miembros que participan en el proyecto. De manera clásica, se suelen afrontar las pruebas de usuario al final del desarrollo, lo que provoca que los errores y las inconsistencias se detecten demasiado tarde, lo que incurre en costes importantes en el proyecto.

En este artículo se ha presentado una solución que mezcla la potencia de la ingeniería guiada por modelos con los paradigmas ágiles para mejorar la realización de las pruebas de aceptación. En este sentido la propuesta ofrece:

1. Que las pruebas de aceptación se realicen de manera independiente del desarrollo del código. Las pruebas se generan desde los requisitos funcionales y por un equipo inde-

pendiente. En este sentido, nuestra propuesta sigue los estándares de referencia.

2. Que los costes de realización de las pruebas se mantengan dentro de los parámetros que tiene el proyecto. El hecho de que las pruebas de aceptación puedan generarse sistemáticamente desde los requisitos con un soporte como NDT-Suite hace que los costes de tener un equipo independiente de pruebas sea asumible por el proyecto.
3. Que las pruebas de aceptación se vayan realizando de una manera "ágil". Las pruebas se van realizando a medida que se va produciendo el desarrollo de los sprint y cualquier error o incidencia se detecta de manera temprana, pudiendo planificarse su resolución en los siguientes sprints.

Con todo ello, la propuesta ofrece un paradigma adecuado en contextos de sistemas como THOT, donde resulta crítica la necesidad de

trabajo de equipos multidisciplinares de una manera efectiva.

Es esencial el poder dar soporte para generar todos estos aspectos. La propuesta, sin el soporte de herramientas para la generación de las pruebas y sin el control efectivo de Redmine conllevaría un coste de gestión demasiado alto y difícil de adecuar.

Actualmente en el proyecto THOT se han presentado los cuatro primeros sprints y la implantación final del proyecto piloto pretende alcanzarse en el mes de Junio de 2014.

Agradecimientos

Esta investigación ha sido parcialmente financiada por el proyecto NDTQ-Framework (TIC-5789) de la Junta de Andalucía, el proyecto MeGus "Mecanismos guiados en etapas tempranas para la mejora de Software" (TIN2013-46928-C3-3-R) del Ministerio de Economía y Competitividad y por el proyecto "THOT. Proyecto de innovación de la gestión documental aplicada a expedientes de contratación de servicios y obras de infraestructuras de transporte" del "Programa Operativo FEDER de Andalucía 2007-2013".

Referencias

- Agile Alliance (2014). <http://www.agilealliance.org/>.
- Canós, J. H.; Letelier, P.; Penadés, M.C. (2003). Metodologías Ágiles en el Desarrollo de Software. // Taller realizado en el marco de las VIII Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2003.
- ENI, Esquema nacional de interoperabilidad. <http://administracionelectronica.gob.es/ctt/verPestanaGeneral.htm?idIniciativa=eni#.UyXVmv15MkM>. (2014).
- Escalona, M.J.; Aragón, G. (2008) NDT. A Model-Driven Approach for Web Requirements. // IEEE Transactions on Software Engineering. 34: 3, 377-390.
- Escalona, M. J.; Gutiérrez, J. J.; Mejías, M.; Aragón, G.; Ramos, I.; Torres, J.; Domínguez-Mayo, F. J. (2011). An overview on test generation from functional requirements. // Journal of Systems and Software. 84:8, 1379-1393. <http://dx.doi.org/10.1016/j.jss.2011.03.051>.
- Grupo IWT2 (2014)., Ingeniería Web y Testing Temprano. <http://www.iwt2.org/>.
- Gutiérrez, J. J.; Nebut, C.; Escalona, M. J.; Mejías, M.; Ramos, I. (2008). Visualization of Use Cases through Automatically Generated Activity Diagrams, Model Driven Engineering Languages and Systems. // Lecture Notes in Computer Science. ISBN: 978-3-540-87874-2, 5301, 83-96.
- IEEE Standard 1008-1987 - IEEE Standard for Software Unit Testing (2014). <http://standards.ieee.org/findstds/standard/1008-1987.html>.
- ISO/IEC/IEEE 29119-1:2013 Software and systems engineering - Software testing - Part 1: Concepts and definitions. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45142. (2014).
- Manifiesto for Agile Software Development (2014). <http://agilemanifesto.org/>.
- Piattini, M. G.; Calvo-Manzano, J. A.; Cervera, J.; Fernandez, L. (2004). Análisis y diseño de aplicaciones informáticas de gestión, una perspectiva de ingeniería del software. Alfaomega. 419-469.
- Ponce, J.; Domínguez-Mayo, F. J.; Escalona, M. J.; Mejías, M.; Pérez, D.; Aragón, G.; Ramos, I. (2010). Pruebas de Aceptación en Sistemas Navegables. // REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software. ISSN 1885-4486. 6:3. 47-55.
- Ponce, J.; Escalona, M. J.; Gómez, A.; Luque, M.; Molina, A. (2010). Definición de una política de pruebas en la gestión cultural: aplicación al desarrollo del proyecto Mosaico. // REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software. ISSN 1885-4486. 6:2, 25-43.
- Proyecto THOT. http://www.aopandalucia.es/inetfiles/agencia_innovacion_estructura/14122012112320.pdf. (2014).
- Pressman, Roger S. (2002). Ingeniería del software: un enfoque práctico. Séptima Edición. McGrawHill. Capítulos 17, 18 y 19.
- Scrum. <https://www.scrum.org>. (2014).

Enviado: 2014-03-31. Segunda versión: 2014-06-24.
Aceptado: 2014-08-11.