

Referencias

- [1] Minoli, D. (1995). *Analyzing Outsourcing: Reengineering Information and Communication Systems*. McGraw-Hill.
- [2] Sturm, R. & Morris, W. (2000). *Foundations of Service Level Management*, SAMS.
- [3] Govekar, M. (2009). Under Pressure: Evolving IT Operations to Best Deliver Business Value, Gartner Symposium, ITxpo 2009, France.
- [4] Verhulst, P.F. (1845). Mathematical Researches into the Law of Population Growth Increase. *Nouveaux Memoires de l'Academie Royale des Sciences et Belles-Lettres de Bruxelles*, 18(1) 1-45.

NDT-Suite, una solución práctica para el uso de NDT

Julián Alberto García García, Daniel Rivero Capellán, María José Escalona Cuaresma, Isabel Ramos Román

Grupo IWT2. Departamento de Lenguajes y Sistemas Informáticos

Universidad de Sevilla

Sevilla - España

{julian.garcia, daniel.rivero}@iwt2.org, mjescalona@us.es, isabel.ramos@lsi.us.es

Abstract: *The importance of software engineering is well established and accepted by the research community. However, it is still common to find reticence in the business world against the discipline. There are plenty of proposals that call for agile methods for developing software. These methods propose very timely measures for the business environment but yet have not been used in large real projects. One of the most important aspects that lead to this situation is the need to implant these methodologies supported by development tools and environments that must be profitable for the business world. This paper presents a proposal framed within the paradigm of model-driven Web Engineering. This proposal has been adapted to provide a methodological environment currently is being used successfully in real environments.*

Resumen: *La importancia de la Ingeniería del Software está ampliamente demostrada y aceptada por la comunidad investigadora. Sin embargo, todavía es frecuente encontrarse reticencias en el mundo empresarial. Hay bastante propuestas que abogan por métodos ágiles y adecuados para el desarrollo de software que proponen medidas muy oportunas para el entorno empresarial pero, que sin embargo, no han sido usadas en grandes proyectos reales. Uno de los aspectos más importantes que llevan a esta situación es la necesidad de acompañar a esas propuestas de herramientas y entornos de desarrollo que realmente sean rentables para el mundo empresarial. Este artículo presenta cómo una propuesta enmarcada dentro del paradigma de Ingeniería Web guiada por modelos se ha adaptado para ofrecer un entorno metodológico que, actualmente, está siendo utilizado de forma satisfactoria en entornos reales.*

Keywords: *Model Driven Web Engineering, Web Requirements, Tools, Practical Experience, NDT*

1. Introducción

A lo largo del tiempo las metodologías web basadas en modelos han ido tomando una relevancia cada vez más importante en la Ingeniería Web, como resultado de esta evolución han surgido distintas propuestas metodológicas que han tratado y tratan de dar respuesta a las dificultades que ofrece el diseño de sistemas hipermedia y aplicaciones web. El éxito o fracaso de estas metodologías se debe a su capacidad para afrontar la realidad del desarrollo de software para la red.

A diferencia de la Ingeniería del Software, la Ingeniería Web ofrece otros retos como son:

Usuarios finales desconocidos. En el desarrollo de aplicaciones web no se debe asumir un rol de usuarios estándar ya que la naturaleza de la web es multicultural además de que cada usuario cuenta con capacidades y tecnología distintas.

Alta disponibilidad y requisitos cambiantes. La propia naturaleza dinámica de la web obliga a que las aplicaciones estén operativas la mayor parte del tiempo y que los cambios en los requisitos o el mantenimiento no supongan un cese de las actividades del usuario.

Ausencia de orientación. De forma general la web es un elemento textual en el que los únicos elementos de navegación son los hipervínculos. Debido a la gran flexibilidad de este mecanismo no es difícil diseñar aplicaciones en las que el usuario se sienta perdido por inconsistencias en la presentación o falta de contexto navegacional.

Por estas razones son necesarias metodologías que se centre en el usuario y que ofrezcan modelos para la fase de requisitos, modelos para el tratamiento de la navegación y modelos de interfaz.

Junto a los problemas a los que dan respuesta las metodologías actuales cabe tener en cuenta la importancia de que estas metodologías estén soportadas por herramientas. Este soporte facilita que los equipos de trabajo se centren en el problema de su dominio de negocio en lugar de preocuparse por los detalles de funcionamiento de la metodología subyacente. Igualmente el soporte de herramientas permite que los usuarios finales puedan validar los modelos generados si las herramientas proporcionan la adaptación de la información de modo que el usuario final no tenga por qué conocer los detalles metodológicos o técnicos del desarrollo para poder validar el trabajo realizado.

El presente artículo pretende ilustrar las herramientas de soporte de la metodología NDT para dar respuesta a las dificultades de desarrollo web junto a un ejemplo de uso y las experiencias derivadas de los proyectos reales realizados con NDT.

2. Trabajos Relacionados

NDT (Navigational Development Techniques) no es la única propuesta existente para la ingeniería basada en modelos para la web. Entre las metodologías más relevantes y que más han influido en NDT se encuentran OOHDM (Object-Oriented Hypermedia Design Method) [2], UWE (UML-Web Engineering) [3] y WebML (Web Modelling Language) [4], todas ellas tienen en común que los metamodelos y modelos están basados en UML y en los principios de orientación a objetos.

OOHDM supuso el punto de partida para las demás metodologías, esta se basa en HDM pero incluye un enfoque orientado a objetos en sus metamodelos. A su vez esta metodología ofrece tres modelos fundamentales que obligan a la separación de conceptos: modelo conceptual, modelo de navegación y modelo de interfaz abstracta.

UWE es una metodología de Ingeniería Web guiada por modelos y sus principales características son: un profile en UML [5] para modelar las fases de desarrollo en particular el modelo de requisitos, modelo de contenidos, modelo de navegación y modelo de procesos. Y soporte tool UWEet (basado en la herramienta UMLet) y MagicUWE (basado en la herramienta MagicDraw).

WebML es la propuesta del departamento de electrónica e informática del politécnico de Milán. Esta propuesta está formada por varios modelos (WebML) y por una herramienta que da soporte a los modelos de WebML llamada WebRatio. La principal característica de esta metodología es su herramienta CASE webRatio basada e integrada con las herramientas de desarrollo eclipse, con capacidades para facilitar el diseño de diagramas BPMN.

En la actualidad podría decirse que NDT compite en el mercado con UWE y WebML por lo que es influido por dichas metodologías y a su vez influye en ellas.

3. Visión general de NDT

NDT (Navigational Development Techniques) [1] es una propuesta metodológica incluida dentro del paradigma MDE (Model-Driven Engineering, ingeniería Guiada por Modelos) que se definió inicialmente para cubrir las necesidades en el desarrollo Web.

NDT comienza con la definición de metamodelos formales para fase de requisitos y la definición de un conjunto de transformaciones definidas en QVT [12] para generar los modelos de fase Análisis. La etapa de ingeniería de requisitos comienza con la definición de los objetivos del sistema. En este sentido, NDT puede entenderse como una metodología guiada por objetivos, según la clasificación en [10].

Una vez definidos los objetivos, NDT propone ir capturando y definiendo los requisitos del sistema y propone dividirlos en diferentes tipos, de manera que puedan tratarse según su tipología. De esta forma, los requisitos se dividen en: requisitos de almacenamiento de información, requisitos de actores, requisitos funcionales, requisitos de interacción y requisitos no funcionales.

La división en tipologías de requisitos sigue la línea que otras propuestas han tenido en el entorno de Ingeniería Web en otras fases, principalmente análisis y diseño, en las que se han separado los modelos para trabajar con cada uno de ellos. Así, por ejemplo, se han definido modelos conceptuales, de navegación, de adaptabilidad, etc. Este tratamiento caracterizado permite tratar de manera particular cada tipología de requisitos y separar los conceptos. De esta forma, cada uno de esos elementos se define de manera formal en NDT mediante un metamodelo en el que cada tipología de esos requisitos, así como sus atributos y las relaciones que se establecen entre ellos, son representados haciendo uso de un diagrama de clases.

En su siguiente fase, el análisis, NDT trabaja de una forma similar. Siguiendo la línea de otras metodologías ampliamente aceptadas como UWE o WebML, NDT propone representar un modelo de contenido, un modelo de navegación y un modelo de interfaz abstracta. Todos ellos, también se encuentran definidos en un conjunto de metamodelos.

La propuesta, analizando tanto los metamodelos de requisitos como los metamodelos de análisis, establece una serie de relaciones entre los artefactos que en ellos aparecen. En base a estas relaciones se definen un conjunto de transformaciones.

Así estaba planteada inicialmente la metodología NDT. Sin embargo, en los últimos años ha evolucionado y ofrece soporte completo para todo el ciclo de vida. Abarca las fases de estudio de viabilidad, requisitos, análisis, diseño, construcción, implementación, así como las fases de mantenimiento y de pruebas durante el desarrollo de software, y además establecen nuevas reglas de transformación. En la figura 1 se muestra cómo a partir de la fase de requisitos es posible obtener los modelos de la fase de análisis y de la fase de pruebas.

Las reglas de transformación de NDT están representadas en la figura 1 mediante el estereotipo «*NDTTransformations*», y una vez que han sido aplicadas, el equipo de analistas puede realizar transformaciones controladas con el objetivo de completar y enriquecer los modelos para obtener el modelo definitivo. Este paso no es automático y requiere la experiencia del analista. Estas transformaciones están representadas en la figura 1 mediante el estereotipo «*NDTSupport*».

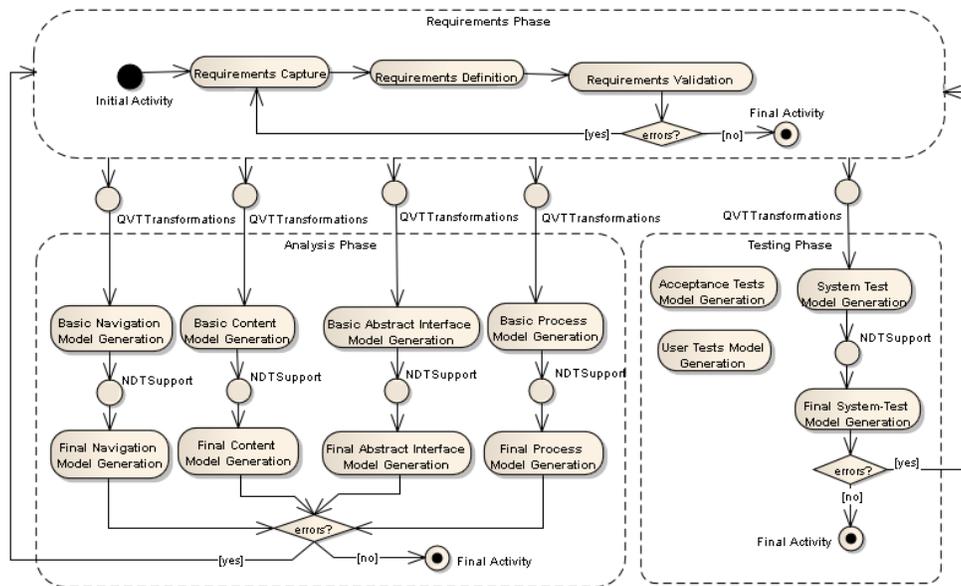


Figura 1. Generación de los modelos de análisis de pruebas desde los requisitos

Por otra parte, NDT es compatible con un conjunto de procesos para llevar a cabo la gestión de proyectos, seguridad y garantía de calidad. Este conjunto de procesos se definen en detalle en el trabajo NDTQ-Framework, presentado en la siguiente sección.

Una ventaja es que NDT se puede utilizar en el entorno empresarial de forma satisfactoria. Hoy en día, un elevado número de empresas en España trabajan con NDT en el desarrollo de software. Esto es posible debido al hecho de que NDT está totalmente apoyado por un conjunto de herramientas libres, agrupadas en NDT-Suite. En las siguientes secciones se detalla el conjunto de herramientas que constituyen esta suite.

4. NDT- Suite

Desde su comienzo, NDT ha sido una propuesta que ha tenido una amplia aplicación en el entorno empresarial. El feedback obtenido tras las aplicaciones prácticas, demostró que la primera herramienta diseñada para dar soporte a NDT, la herramienta NDT-Tool, no resultaba una solución óptima para el trabajo en proyectos reales de gran envergadura y heterogéneos debido principalmente a que NDT-Tool no era flexible; en [6] se justifica esta falta de flexibilidad de NDT-Tool. Por estos motivos, surge la necesidad de desarrollar la nueva suite de herramientas NDT: *NDT-Suite*.

Para desarrollar NDT-Suite lo primero que se hizo fue hacer una extensión de la propia metodología. Tomando las ideas de NDT y siguiendo las premisas marcadas por la metodología Métrica v3¹, UWE y OOHDM, se hizo una extensión para abordar todo el ciclo de vida. Esta extensión ha sido publicada en [[7]].

De esta forma, y aprovechando las premisas de estos entornos, suficientemente afianzados tanto en la Ingeniería del Software como la Ingeniería Web, y las ideas esenciales de NDT, se ha definido un entorno de trabajo compuesto por las fases de Requisitos, Análisis, Diseño, Construcción e Implantación, Pruebas y Mantenimiento. El hecho de que la elección haya sido Métrica se ha debido a que es la metodología de

¹ www.map.es

referencia en las Administraciones Públicas españolas. UWE fue seleccionado porque es una metodología Web fundamentada sobre UML, aspecto de gran interés debido a que es el lenguaje soportado por Métrica. Y OOHDM fue elegido por ser la primera metodología para la Web, cuyos modelos han sido adaptados y aprobados por toda la comunidad investigadora. La fusión realizada se basa en definir un proceso, similar al de Métrica pero haciendo uso de los modelos de UML y de las extensiones que NDT realiza de ellos, así como de sus procesos de ingeniería guiada por modelos.

A lo largo de este apartado se presentan todas las herramientas que actualmente componen NDT-Suite y otras que serán liberadas en breve.

4.1. NDT-Profile

El primer trabajo que dio inicio a NDT-Suite fue seleccionar un entorno de herramientas adecuado para trabajar. Tras diferentes estudios se optó por Enterprise Architect [8] (EA en adelante). Esta herramienta, a pesar de no ser de software libre, ofrece un soporte adecuado y su precio es bastante competitivo. Además, ofrece varias ventajas importantes como la posibilidad de definir perfiles, herramientas para la gestión de documentación, etc. que han resultado de gran interés para nuestro trabajo.

La ampliación de la metodología NDT supuso la definición de nuevos metamodelos y transformaciones para todas las fases del ciclo de vida software. Para cada uno de los metamodelos de NDT se definió un profile en la herramienta EA. Este entorno es lo que se conoce como *NDT-Profile*. Así, por ejemplo, en la figura 2 se presenta una captura del entorno de trabajo en el que puede verse remarcado a la izquierda los distintos toolbox de NDT.

El uso de NDT-Profile ofrece la posibilidad de disponer de todos los artefactos de NDT de una manera sencilla, puesto que están integrados en la propia herramienta pero, además, también permite utilizar todos los modelos de UML e integrarlos fácilmente en la metodología.

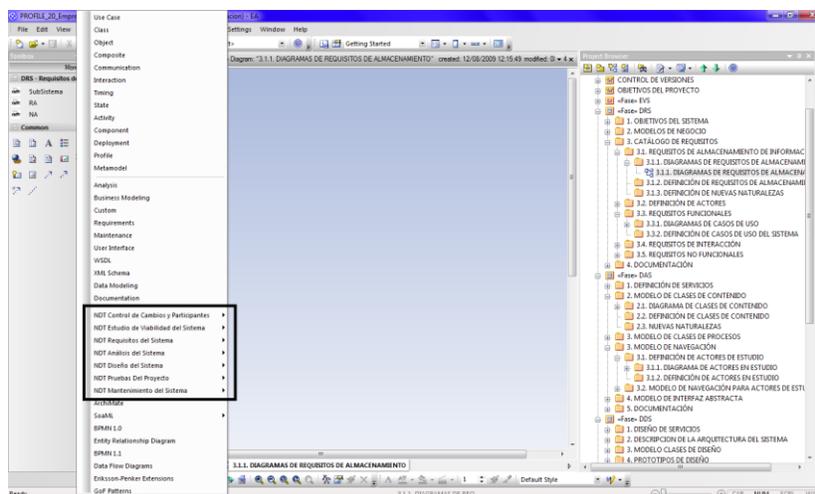


Figura 2. Interfaz principal de NDT-Profile. Toolbox y navegador del proyecto

4.2. NDT-Quality

El uso de NDT-Profile revela un problema importante y es la flexibilidad de trabajo que esta herramienta permite. Con NDT-Profile es posible infringir fácilmente las reglas y relaciones definidas en NDT, las cuales, son esenciales para trabajar con el entorno MDE de la propuesta. Por ello, se ha desarrollado otra herramienta dentro de NDT-Suite que se denomina NDT-Quality.

NDT-Quality, figura 3, es una herramienta que automatiza la revisión metodológica de un proyecto desarrollado con NDT-Profile. Se encarga de revisar tanto la calidad de un entregable en cuanto al uso de la metodología NDT en cada una de las fases del ciclo de vida software, como la trazabilidad de las reglas MDE que define NDT entre las distintas fases del ciclo de vida. Además, para facilitar el trabajo en proyectos que siguen ciclos de vida iterativos, la herramienta permite mediante una ventana de conformación seleccionar qué reglas concretas se desean verificar, figura 4.

Una vez realizada la revisión, la herramienta proporciona, para cada fase del ciclo de vida software revisada, un listado con los errores o inconsistencias metodológicas detectadas, figura 5. Estos errores se tipifican como leves, graves, o críticos. Entre los aspectos que son revisados por la herramienta se encuentran por ejemplo, completitud en definiciones, definiciones erróneas de restricciones, detección de ciclos en diagramas de clases, etc.

Para facilitar la creación de informes, la herramienta permite exportar en un documento en diferentes formatos el listado de inconsistencias encontradas durante la revisión.

Finalmente, como valor añadido, la herramienta no sólo se limita a resaltar los errores detectados durante la revisión, sino que también proporciona para cada uno de ellos una guía para resolverlo.

Como conclusión final, el uso de NDT-Quality permite garantizar la calidad de cualquier proyecto que siga la metodología NDT.

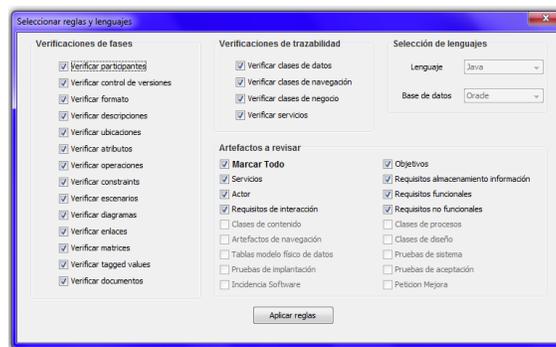


Figura 3. Ventana de configuración de NDT-Quality

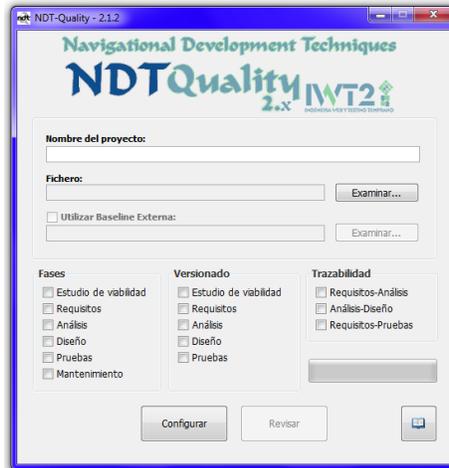


Figura 4. Ventana de NDT-Quality

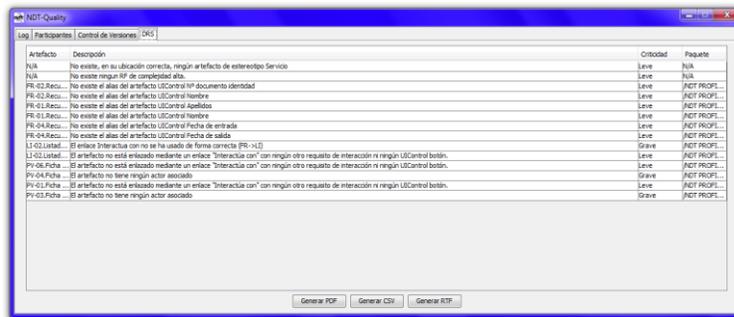


Figura 5. Ventana de informe de NDT-Quality

4.3. NDT-Driver

Inicialmente, NDT sólo definía la transformación del modelo de requisitos del proyecto en los modelos de la fase de análisis. A raíz de la ampliación de NDT al resto de fases del ciclo de vida, se tuvieron que definir nuevas transformaciones. Por este motivo, se hizo necesario incorporar esta nueva funcionalidad en la suite de NDT mediante una nueva herramienta: NDT-Driver.

NDT-Driver, figura 3, es una de las principales herramientas de soporte de la metodología NDT. Implementa un conjunto de procedimientos automáticos para llevar a cabo cada una de las transformaciones QVT definidas en NDT. Es capaz de generar los modelos de la fase de Análisis desde el modelo de la fase de Requisitos, los modelos de la fase de Diseño desde los modelos de Análisis, y el modelo de pruebas de sistema de la fase de Pruebas desde el modelo de Requisitos. Además, NDT-Driver permite obtener el modelo Requisitos a partir de los requisitos capturados durante la fase de Estudio de Viabilidad del proyecto. Además, para facilitar el trabajo en proyectos que siguen ciclos de vida iterativos, para cada transformación, NDT-Driver le permite al usuario seleccionar el modelo concreto a generar. En la figura 6 se muestra una ventana de configuración en la que aparecen los modelos disponibles para la transformación de Requisitos a Análisis.

La aplicación de alguna de las transformaciones que define NDT proporciona, sin duda, una posición aventajada al analista a la hora de desarrollar los distintos modelos de una determinada fase del ciclo de vida. Sin embargo, estos modelos generados de forma automática deben ser mejorados por el analista. Durante este

proceso de mejora es posible que se detecten requisitos del cliente que no han sido considerados en el modelo origen de la transformación, lo cual, conllevaría retocar el modelo origen y volver a generar el modelo en cuestión, lo cual además conllevaría la pérdida de todo el trabajo realizado por parte del analista en cuanto a la mejora del modelo generado.

Como solución a este aspecto, NDT-Driver permite dos modos de transformación: Reconstrucción o Actualización. El modo Reconstrucción consiste en volver a generar un modelo desde cero partiendo del modelo origen y descartando el modelo generado, si éste ha sido previamente generado. Mientras que gracias al modo Actualización, sólo se transforman aquellos elementos del modelo original que han sido modificados.

Por ejemplo, si todos los requisitos de almacenamiento han sido ya definidos en la fase de Requisitos, es posible generar el modelo conceptual de la fase de Análisis. Si llegados a este punto, se detectara que algún requisito de almacenamiento no se ha definido según las necesidades del cliente, no es necesario volver a generar de nuevo el modelo conceptual; NDT-Driver permite actualizarlo.

Como conclusión final, el uso de NDT-Driver permite reducir cuantitativamente el tiempo de desarrollo de cualquier proyecto que siga la metodología NDT.



Figura 6. Interfaz de NDT- Driver - Ventana de NDT Driver

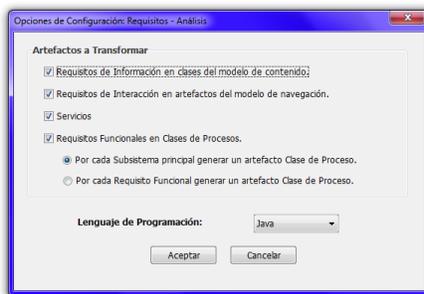


Figura 7. Interfaz de NDT- Driver - Ventana de configuración NDT Driver

4.4. NDT-Report

Toda la información descrita y desarrollada en un proyecto realizado en base a la herramienta NDT-Profile, aunque práctica y funcional, no es amigable a la hora de presentar a usuarios y clientes. Por este motivo, se desarrolló y liberó en 2008 la primera versión de una herramienta de generación documental para NDT denominada *NDT-Report*. Esta herramienta contemplaba la generación de un documento con todos los

requisitos recogidos en la fase de Requisitos y los modelos definidos en la fase de Análisis, generando el documento de análisis.

Debido a la ampliación que ha experimentado la metodología NDT en los últimos años, NDT-Report ha evolucionado con el objetivo de cubrir las necesidades documentales de todas las fases del ciclo de vida.

Actualmente, y aprovechando la potencia de EA para la generación de documentos basada en plantillas personalizadas, NDT-Report está constituido por un amplio conjunto de plantillas integradas dentro de la herramienta NDT-Profile. Con NDT-Report puede generar fácilmente un documento para cada fase del ciclo de vida de NDT: desde la fase de Requisitos hasta la de Mantenimiento, pasando por la fase de Análisis, la de Diseño y la fase de Pruebas del proyecto.

4.5. NDT-Glossary

Durante el desarrollo de un sistema de información, orientado o no a la Web, pueden surgir problemas de terminología debido principalmente a que en el proyecto participan grupos de personas muy heterogéneos y con diferente nivel de conocimiento del sistema a desarrollar: clientes, usuarios finales, jefes de proyecto, diseñadores, ingenieros, etc. Este problema se acrecienta en proyectos de sistemas de información Web en los que los equipos de trabajo son multidisciplinares. Para intentar paliar este problema NDT incluye en su suite una herramienta para la construcción de un glosario terminológico, denominada *NDT-Glossary*.

NDT-Glossary implementa un procedimiento automático que, a partir del análisis de los requisitos de un proyecto de un sistema de información, orientado o no a la Web, y que está realizado en base a la herramienta NDT-Profile, genera la primera versión del glosario terminológico de dicho proyecto. El conjunto de términos, o conceptos, recopilados pasarán a formar parte del glosario terminológico del proyecto en cuestión.

Cada entrada del glosario, correspondiente a un término, contendrá una descripción y un nombre, siendo este último único en el glosario.

Además de recopilar los conceptos del proyecto, NDT-Glossary ofrece la posibilidad de crear un documento en diferentes formatos (PDF o RTF), dónde se recoge de manera ordenada y estructurada toda la información del glosario de términos generado.

4.6. NDT-Prototypes

La única forma de que NDT pueda ser una realidad empresarial pasa por ofrecer herramientas que den soporte al desarrollo de sistemas de información Web. En esta línea de actuación se engloba *NDT-Prototypes*. Esta nueva herramienta de NDT-Suite ha sido liberada recientemente y está proceso de mejora y ampliación de funcionalidad.

Implementa un procedimiento que genera de manera totalmente automática un conjunto de prototipos XHTML a partir de los modelos de navegación descritos en la fase de Análisis de un proyecto desarrollado con la herramienta NDT-Profile. Estos prototipos proporcionan al equipo de desarrollo un punto de partida para poder llevar a cabo la construcción del sistema.

A efectos prácticos, NDT-Prototypes no sólo permite disminuir el tiempo empleado en la construcción de una aplicación Web sino que también proporcionan una herramienta útil para la validación de requisitos con usuarios y clientes.

4.7. Otras Herramientas

En esta sección se describen otras herramientas que se consideran relevantes para proporcionar al lector de una visión completa de NDT-Suite: NDT-Access, NDT-Counter y NDTQ-Framework.

En primer lugar, *NDT-Access* es una herramienta que tiene como objetivo agilizar el desarrollo de nuevas herramientas dentro del paquete NDT-Suite. NDT-Access supone el pilar maestro sobre el que se sustentan todas las herramientas que han sido, y serán, desarrolladas para proporcionar soporte a la metodología NDT para su uso en entornos prácticos. Proporciona una interfaz de comunicación totalmente transparente para llevar a cabo consultas, creaciones, modificaciones y actualizaciones de los datos del proyecto. La implementación de esta herramienta se ha llevado a cabo haciendo uso de la API que proporciona Deiser, empresa distribuidora de Enterprise Architect. Además de transparente, NDT-Access proporciona una interfaz independiente del sistema gestor de bases de datos que se utilice para almacenar los artefactos de NDT.

En segundo lugar se presenta la herramienta *NDT-Counter*. Cada una de las herramientas de NDT-Suite descritas anteriormente, están orientadas a cubrir una determinada necesidad del ciclo de vida software que define NDT. Sin embargo, ninguna de ellas proporciona soporte a la etapa de gestión del proyecto, y más concretamente, soporte a la tarea de estimación del coste que va a suponer el desarrollo de un proyecto software. Para cubrir esta necesidad se desarrolla una nueva herramienta, la cual aún no ha sido liberada, dentro de la suite de NDT: *NDT-Counter*. La técnica de estimación que implementa esta herramienta está basada en la técnica de puntos de casos de uso.

Por último, se presenta brevemente *NDT-Q-Framework*². NDT-Q-Framework no es una herramienta software en sí misma, sino un marco de trabajo para el desarrollo de software de calidad. Ofrece un entorno de trabajo sostenible y ágil adecuado para tanto el desarrollo de aplicaciones software en entornos de las administraciones públicas y empresas privadas. En este marco de trabajo se plantea un desarrollo orientado por procesos y que contempla, no sólo las fases de desarrollo, sino también los aspectos de calidad, gestión, pruebas y mantenimiento. Este marco de trabajo contempla la definición de los procesos, la definición de sus políticas de orquestación y de las herramientas necesarias para su ejecución.

4.8. Ejemplo de uso de NDT-Suite

En este apartado se describirá cuál debería ser la secuencia de trabajo para lograr generar el modelo conceptual final de la fase de análisis. Para este supuesto, supongamos un sistema que permite la gestión de las reservas y estancias de un hotel.

En primer lugar, el equipo de analistas debe utilizar NDT-Profile para comenzar a registrar y catalogar los distintos requisitos del sistema a desarrollar. Por ejemplo, entre los requisitos de almacenamiento de

² Más información sobre NDTQ-Framework en: <http://www.iwt2.org/iwt2/ndt-qframework.php>

NDT-Profile es utilizado por un gran número de proyectos, tanto pequeños (unos 20 o 25 requisitos) hasta proyectos muy grandes (más de 2000 requisitos). Proyectos abordados por equipos heterogéneos o simples y se encuentra en continua mejora.

A medida que se extendía el uso de NDT-Profile y que recibíamos y analizábamos los problemas con la se encontraban los equipos, se comenzó a abordar el desarrollo de las otras herramientas. NDT-Quality resultó un hito importante puesto que permitía, de una manera sencilla, tanto al equipo de desarrollo como a los jefes de equipo controlar si NDT y, por supuesto el profile, se estaban usando de manera correcta.

Aunque no es interés de este trabajo mostrar estadísticas numéricas, sirva de ejemplo que en los inicios de Diraya Especializada, por ejemplo, revisar la calidad de los trabajos realizados por una de sus seis empresas podría tomar más de 1 semana. Actualmente, con NDT-Quality, no se tarda más de 2 minutos. Además, como los equipos disponen de la herramienta pueden validar ellos mismos sus trabajos.

La siguiente herramienta fue NDT-Driver y esta ha sido sin duda una de las más aceptadas en el mundo empresarial. A pesar de que desde el mundo académico se haya demostrado la necesidad de los análisis, la documentación, etc. para los equipos de desarrollo la generación de la documentación sigue siendo una tarea poco valorada y que suele entenderse más como “algo que hay cumplir y que quita tiempo”. NDT-Driver permite automatizar en gran medida todo este trabajo. Así, obtener una primera propuesta de análisis desde una ingeniería de requisitos es tan rápido como un clic de ratón.

A estas herramientas también le han llegado muchas críticas. Críticas que intentamos analizar y solventar. Por ejemplo, una de las más importantes ha llegado desde la Consejería de Cultura y la Consejería de Innovación, Ciencia y Empresa en un proyecto conjunto. Ellos siguen un ciclo de vida iterativo e incremental, en lugar de secuencial. Así, por ejemplo, si en la primera iteración se definían los requisitos, usando NDT-Driver se generaba el análisis básico y ellos evolucionaban hacia un análisis final, cuando en la siguiente iteración incrementaban los requisitos y volvían a usar NDT-Driver, este destruía el trabajo de análisis realizado en la fase anterior.

Para salvar este problema, se ha enriquecido a NDT-Driver con una opción de generación en bloques adaptado especialmente para este tipo de ciclo de vida.

En la actualidad, para todos los proyectos en los que colaboramos como equipo, se llevan una serie de indicadores e inspectores que nos permiten medir o evaluar el uso de todo este entorno. Además, nos ofrecen mucha información relevante para trabajos futuros.

Resulta muy interesante ver cómo, intentando satisfacer a los equipos empresariales de desarrollo con entornos adecuados para ellos que cubran sus necesidades existe una aceptación y una mejor aplicación de las metodologías de trabajo.

6. Conclusiones y trabajos futuros

Este trabajo ha analizado la evolución práctica de NDT y cómo la metodología ha tenido que adaptarse ofreciendo un entorno de herramientas, llamado NDT-Suite, que ofrece una manera sistemática y que vela por la calidad de los resultados.

El trabajo ha presentado una visión general de NDT y una presentación de las herramientas que componen el entorno NDT-Suite. Con un ejemplo práctico se ha tratado de ilustrar cómo se utilizarían estas herramientas y también se ha justificado con un breve recorrido de las experiencias prácticas cómo ha ido surgiendo la necesidad de su existencia.

Respecto a los trabajos futuros sin duda son muchos. La mejora continua de estas herramientas que nos llegan desde los proyectos reales en los que se están utilizando es una fuente de modificaciones, adaptaciones, evoluciones, etc. constante que nos está llevando a ir depurando cada día nuestras propuestas.

Por otra parte, la fase de implementación que incluye construcción e implantación, queda aislada. En este sentido se ha realizado un trabajo en colaboración con la empresa Sun Microsystem y que, actualmente sólo se está usando en uno de los proyectos, el proyecto Diraya Especializada. Esta herramienta, basándose en el análisis del código Java y de la base de datos de Enterprise en la fase de diseño, se encarga de validar que el diseño se corresponde con el código. Esta herramienta, que ha recibido el nombre de CADI es otro de los puntos futuros de trabajo, y que nos están abriendo un trabajo importante para asegurar la calidad y la trazabilidad de los proyectos.

La búsqueda de inspectores y métricas que se obtengan de la manera más sistemática posible es otra línea abierta. Ofrecer indicadores que permitan validar la corrección de los entregables, la evaluación de la calidad de lo que se entrega, etc. es un punto esencial que es necesario para los jefes de equipo y los directores. En la actualidad se han definido un conjunto de inspectores que se llevan midiendo sobre los proyectos desde hace más de un año. Sin embargo, es necesario automatizar la consecución de estos inspectores puesto que actualmente es un trabajo demasiado manual. En la actualidad se están buscando alternativas para que NDT-Quality incluya esos medidores y una vez obtenga la evaluación indique esas medidas para los proyectos.

Agradecimiento

En este trabajo de investigación ha sido soportado por el proyecto Tempros (TIN2010-20057-C03-02), por RED CaSA (TIN2010-12312-E) del Ministerio de Ciencia y Educación, España, y por el proyecto NDTQ-Framework de la Junta de Andalucía, España (TIC-5789).

Referencias

- [1] Escalona, M.J. & G. Aragón, G. (2008). NDT. A model-driven approach for web requirements. *IEEE Transaction on Software Engineering*, 34(3) 377-390.
- [2] Rossi, G. (1996). An Object-Oriented Method for Designing Hypermedia Applications. PHD Thesis, University of PUC-Rio. Rio de Janeiro. Brazil.