



Microsoft Test Manager

Planificando nuestras pruebas

Con la llegada de Visual Studio 2010, hemos recibido también un nuevo producto: Microsoft Test Manager. Hasta ahora no existía ninguna herramienta completa para pruebas funcionales dentro de los productos Microsoft. Con esta nueva herramienta, podremos incorporar el equipo de pruebas dentro de nuestro ciclo de vida de desarrollo con Visual Studio 2010 y Team Foundation Server 2010.

En multitud de equipos de desarrollo, el equipo de pruebas es ese grupo apartado del resto y armado con documentos Excel o Word en los que detallan todo lo que tiene que hacer la aplicación. Documentos que en ocasiones incluso solo ellos manejan, y de los que no se tiene un extenso conocimiento por parte del resto del equipo. Por otra parte, cada vez hay más equipos de desarrollo que adoptan metodologías ágiles, y con ello uno de sus principales conceptos: los equipos multidisciplinares. Esto no quiere decir que todo el mundo sepa hacer de todo, sino que todo el mundo está dentro del equipo y comparte la misma información con total transparencia.

Con **Microsoft Test Manager** (MTM), podemos ayudarnos para conseguir que todo el ciclo de pruebas sea parte integral del proceso de desarrollo en equipo. Se trata de un producto nuevo, del que por desgracia aún no hemos hablado mucho por aquí (asumo mi parte de culpa ante los lectores), por lo que en éste y próximos artículos iremos presentando poco a poco las muchas nuevas funcionalidades que tenemos a nuestra disposición a través de esta herramienta.

Lo primero de todo: ¿qué necesitamos para poder utilizar MTM? En primer lugar, un Team Foundation Server 2010, ya que todos los casos de test se almacenan como *work items* de TFS. Como herramienta de cliente, necesitaremos o bien Visual Studio 2010 Test Professional, o Visual Studio 2010 Ultimate.



Luis Fraile

Microsoft MVP de ALM y responsable de la Microsoft ALM Division de Testhouse

Creación del plan

Existen varios tipos de pruebas a realizar en relación con un proyecto: pruebas de aceptación de usuario, pruebas de definición de cuándo algo está *hecho* (“done”), pruebas de regresión, pruebas de tipo *end-to-end*, etc. Con MTM, podemos agrupar nuestros casos de prueba mediante **planes de pruebas**. Un plan de pruebas se puede corresponder, por ejemplo, con el plan de pruebas de una iteración o el plan previo a una determinada *Release*.

Cuando nos conectemos por primera vez a MTM, después de seleccionar el servidor de TFS y el proyecto, podremos crear nuestro primer plan (figura 1), pulsando el botón “Add”. Para esta ocasión no vamos a entrar en más detalles, y simplemente crearemos un plan básico; en general, dentro de las propiedades de un plan podemos especificar más valores de configuración del mismo.

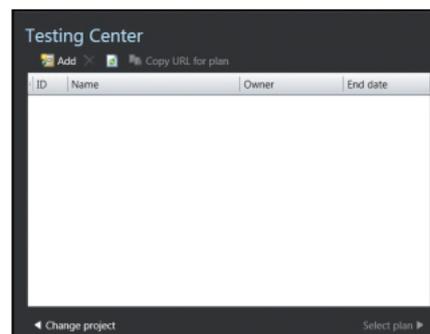


Figura 1

Una vez creado el plan, pulsamos en “Select plan”, con lo que entraremos en la pestaña “Plan” dentro de MTM (figura 2).

de pruebas. Con MTM podemos crear *suites* de tres tipos diferentes; vamos a ver cuáles son y cómo crearlas.

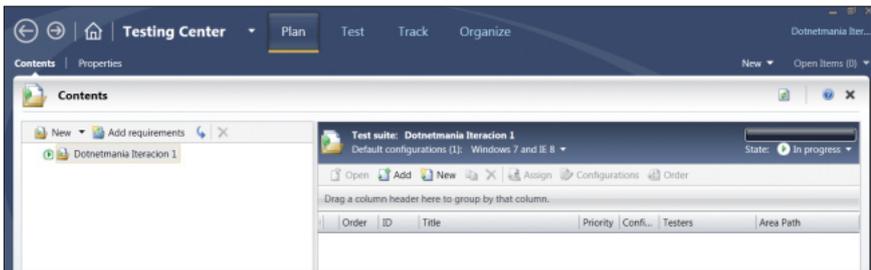


Figura 2

Para un proyecto concreto de TFS podremos tener todos los planes de pruebas que necesitemos, así como todos los casos de pruebas, que podrán estar asignados a varios planes a la vez. Pensad por ejemplo en una estructura en la que tengamos un plan de pruebas por cada iteración, y otro plan de pruebas para comprobar cada *release* que vayamos a hacer. Dentro de este último, podremos tener los casos de prueba más críticos del sistema, pruebas *end-to-end*, o pruebas de regresión que hayamos ido creando y ejecutando durante las iteraciones que componen esa *release*.

En la figura 2 podemos observar también un hipervínculo llamado “Properties” en la parte superior. Si pulsamos en él, podremos entrar en la configuración del plan de pruebas (figura 3); aquí podremos definir datos como qué resultados de compilación de TFS vamos a usar para este plan de pruebas, de modo que las personas responsables de las pruebas sepan cuándo tienen nuevos resultados (binarios) para probar; qué entornos que vamos a probar como sistemas operativos o navegadores; los entornos físicos o virtuales disponibles para el plan; e incluso qué tipo de datos de diagnóstico queremos obtener (vídeo, Intellitrace, visor de eventos, etc.).

Creación de suites de pruebas

Dentro de un plan de pruebas, vamos a organizar los casos de prueba en *suites*

En base a una historia de usuario o requerimiento

Recordemos que en TFS todo son *work items*, y las historias de usuario o requerimientos también lo son. Con la llegada de la versión 2010, se introdujo la capacidad de definir jerarquías de *work items*, que para este caso son ideales. De lo que se trata con este tipo de *suite* es de definir qué pruebas son necesarias para cada una de las historias de usuario a entregar en una iteración. Hay que recordar que, usemos metodologías ágiles o no, tenemos que tener claro cuando algo está completamente implementado y cumple con el objetivo de negocio escogido.

Para crear una suite de pruebas para una historia de usuario, desde la vista “Plan” de MTM seleccionaremos la opción “Add requirements” (figura 4).

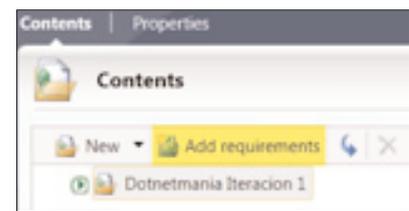


Figura 4

A continuación, se nos preguntará qué consulta de *work items* queremos ejecutar para seleccionar la historia de usuario que queremos añadir a nuestro plan. Esta consulta se basa en el nuevo concepto de categorías de *work items*, fil-

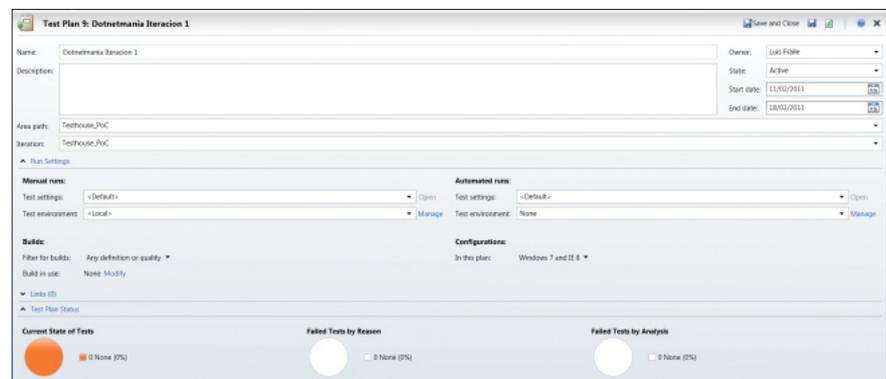


Figura 3

trando por la categoría de Requerimientos, a la que pertenecen tanto los requerimientos para la plantilla de CMMI, como las historias de usuario para la plantilla de MSF Agile 5.0.

Una vez añadimos el requerimiento, y creamos o le añadimos casos de prueba existentes, esto se traduce en un enlace de *work items* con tipo **Tested by**. Esto nos proporciona una ventaja a la hora de obtener informes, ya que podremos saber qué historias de usuario están cubiertas por casos de prueba, cuáles han pasado, cuáles no, y, antes de entregar los resultados de una iteración o *release* saber cuál es el estado de sus pruebas.

Manuales

Este es el tipo de *suite* más simple que podemos crear. En ella agregaremos manualmente todos los casos de prueba que queramos que formen parte de la *sui-*

te. Podemos utilizar una *suite* de este tipo, por ejemplo, para agregar manualmente todos los casos de prueba que definamos para escenarios *end-to-end*, agregando casos de prueba de varias historias de usuario. O bien una *suite* de pruebas de regresión de defectos localizados y reparados anteriormente, y que pertenecen a multitud de historias de usuario y otras *suites* de pruebas.

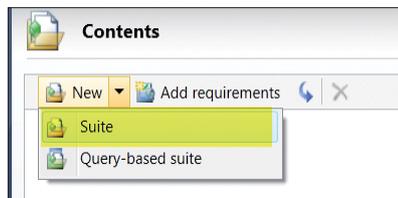


Figura 5

Para crear una suite de pruebas manual, desde la pestaña “Plan” seleccionamos “New” | “Suite” (figura 5) y simplemente le damos un nombre a nuestra nueva *suite*.

Una ventaja adicional de este tipo de *suites* consiste en que podemos crear, por debajo de ellas, sub-*suites* de cualquiera de los tres tipos de *suites* disponibles, lo que nos permitirá organizar aún mejor nuestras *suites* de pruebas.

En base a una consulta de work items

En este tipo de *suite* de pruebas, nos basamos en una consulta de *work items* creada a medida por nosotros. Este escenario es útil, por ejemplo, para crear *suites* de pruebas del tipo “todos los casos de prueba de la iteración 2 con prioridad 2”, o “todos los casos de prueba marcados como críticos del área de frontal web”.

Estas *suites* la crearemos desde el mismo menú que las manuales, pero seleccionando la opción “Query based suite” (figura 6).

Lo siguiente que obtendremos, al igual que en el caso anterior, es una pantalla donde crear la consulta que define qué *work items* serán añadidos a la *suite*. Una de las grandes ventajas de este tipo de *suite* es que son, podríamos decir, *dinámicas*; es decir, que si seleccionamos como consulta “todos los casos de prueba con prioridad 2”, cual-

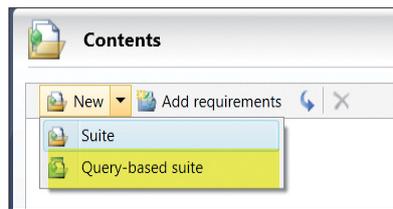


Figura 6

quier nuevo caso de prueba de prioridad 2 que se agregue al proyecto pasará a formar parte de esta *suite* automáticamente.

Al igual que en el caso de las *suites* manuales, para este tipo de *suites* podremos crear sub-*suites* para organizar mejor nuestros casos de prueba.

Agregando caso de prueba

Una vez que ya tenemos nuestro plan de pruebas definido, es el momento de agregar casos de prueba.

Esto lo hacemos también desde la sección “Plan”, a la derecha de las *suites*. Seleccionando cualquiera de las *suites* de pruebas disponibles, tendremos los botones “Add” y “New”, que nos permitirán tanto agregar casos de prueba ya existentes como crear nuevos casos de prueba (figura 7).

La única excepción aquí son las *suites* basadas en una consulta de *work items*. En estas *suites*, dada su naturaleza, no podemos añadir o crear nuevos casos de pruebas; simplemente, cuando se cree un nuevo test que cumpla las condiciones corres-

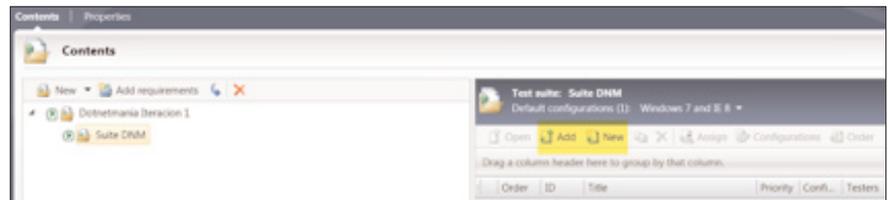


Figura 7

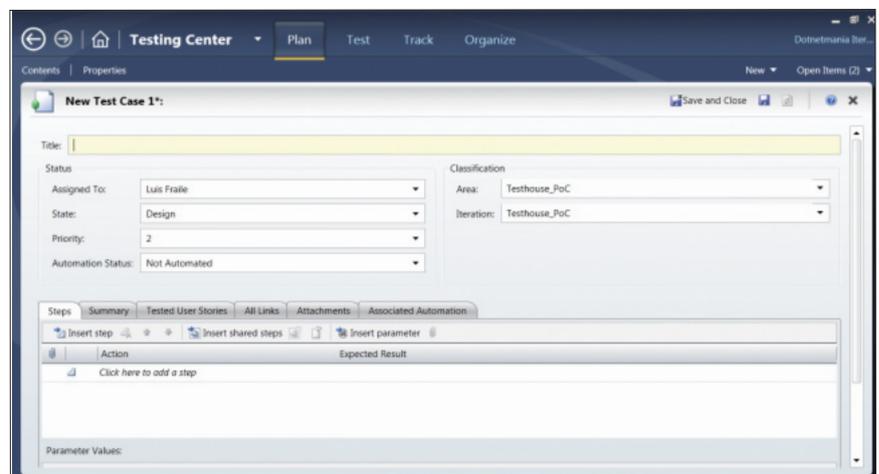


Figura 8

Hay que reseñar que las *suites* de pruebas solo tienen existencia en MTM; no es un concepto que vayamos a ver reflejado, por ejemplo, en Team Explorer (únicamente las *suites* del primer tipo, mediante los enlaces de *work items*, se verán reflejados ahí). No obstante, toda esta información está en la base de datos de TFS.

pondientes, éste se agregará automáticamente.

Recordad que los casos de prueba, a fin de cuentas, no son más que *work items*, aunque solo pueden ser completamente editados desde MTM (figura 8), ya que la parte de dar de alta los casos de prueba solo está soportada directamente en MTM y no a través de Visual Studio. Pero sí podremos consultarlos desde cualquiera de los clien-

tes de TFS, como Team Explorer o el cliente web.

Conclusiones

Espero que este primer artículo acerca de Microsoft Test Manager os haya servido como introducción a esta completa herramienta. Sin duda, lo que hemos mostrado aquí no constituye la parte más espectacular de MTM, como pueden ser la ejecución automatizada de pruebas o la gra-

bación de sesiones y recolección de datos (por ejemplo, de Intellitrace). Pero sí que hemos sentado las bases para llevar a cabo una primera, y consistente planificación de nuestros esfuerzos de pruebas, de modo que los tests queden accesibles a todos los miembros del equipo, y alineados con los objetivos de las historias de usuario y nuestros planes de *release*. En próximas entregas, continuaremos presentando las principales posibilidades que nos ofrece MTM. 

Referencias

[1] Recursos para desarrolladores en MSDN: <http://msdn.microsoft.com/es-ES/>.



Testhouse

Testhouse es una compañía internacional especializada en la consultoría de garantía de calidad, pruebas de aplicaciones y formación. Con sede central en Londres, Reino Unido, Testhouse opera en América del Norte, en el centro y sur de Europa, en Oriente Medio y en Asia y Pacífico. Desde el año 2000, Testhouse se dedica a "testear software", detectando errores y minimizando riesgos para multitud de empresas en diferentes sectores. Nuestra web está en: <http://www.es.testhouse.net>.

Microsoft®

**WEB
CAMPS**



Make Web Not War

Queremos invitarte a que participes en nuestro evento "Web Camps" durante los próximos días en diferentes ciudades. De la mano de expertos de Microsoft podrás aprender y construir en la plataforma Web de Microsoft, así como desarrollar productivamente las nuevas tecnologías web.

- **Web Camps Barcelona**
Día: 22 de marzo
Lugar: Barcelona Activa
- **Web Camps Sevilla**
Día: 31 de marzo
Lugar: Universidad de Sevilla
- **Web Camps Valencia**
Día: 24 de marzo
Lugar: Bolsa de Valencia
- **Web Camps Madrid**
Día: 5 de abril
Lugar: E.T.S.I.T- UPM
- **Web Camps Bilbao**
Día: 29 de marzo
Lugar: TISA- Parque Tecnológico

Los asistentes obtendrán
50% de dto.
en la compra del libro¹
"Programación de Silverlight 4"
y en los cursos
de Aula Vulcan²

No esperes más

Más información
y Registro → → →

www.microsoft.es/makeweb

¹ descuento no acumulable a otras ofertas, para envíos a España.

² descuento no acumulable a otras ofertas, para cursos presenciales de más de 12 horas en inscripciones hasta el 30 de junio de 2011.