



MINISTERIO
DE ECONOMÍA Y
HACIENDA

SUBSECRETARÍA

SUBDIRECCIÓN GENERAL DE
TECNOLOGÍAS DE LA
INFORMACIÓN Y DE LAS
COMUNICACIONES

COMUNICACIÓN TECNIMAP 2010

Mejora de la calidad en el desarrollo software con TFS y Scrum

Santiago Vélez Fraga

Subdirección General de Tecnologías de Información y de las Comunicaciones

Subsecretaría

Ministerio de Economía y Hacienda

CORREO ELECTRÓNICO:

sgtic@subsecretaria.minhac.es

ALCALÁ 5
28070 MADRID
TEL: 91 595 81 46
FAX: 91 595 85 33



ÍNDICE DE LA COMUNICACIÓN

PÁGINA

1.	Temario en el que se adscribe la comunicación.....	2
2.	Palabras clave	2
3.	Resumen de la Comunicación.....	2
4.	Introducción y objetivos. La calidad en el desarrollo software	3
5.	Implantación de Microsoft Team Foundation Server y plantilla de Scrum	5
5.1.	Buenas prácticas en la gestión del ciclo de vida de las aplicaciones. Aspectos a cubrir	5
5.2.	Trabajo en equipo con TFS, y aplicación de una metodología de desarrollo Scrum....	9
5.3.	Conclusiones y planificación de la implantación de Team Foundation Server en la SGTIC del Ministerio de Economía y Hacienda.....	11

1. Temario en el que se adscribe la comunicación

- Servicios para los usuarios, en particular en el punto de "Calidad del servicio".
- Eficiencia y sostenibilidad, en el punto de "Reingeniería de procesos".

2. Palabras clave

Calidad del desarrollo software, oficina de proyecto, metodologías ágiles, Scrum, Team Foundation Server.

3. Resumen de la Comunicación

Calidad en el desarrollo software para mejorar el servicio al ciudadano.

Implantación de metodologías ágiles (Scrum) y herramientas de control del ciclo de vida del software (TFS), y su impacto positivo en el servicio al ciudadano

En el área de Desarrollo de la SGTIC del Ministerio de Economía y Hacienda se ha planteado un proyecto para la mejora del proceso de desarrollo de los productos software de la Subdirección con el objetivo final de dar un mejor servicio.

- Un software desarrollado con más calidad cumple las expectativas de sus usuarios y de los ciudadanos, en funcionalidad y coste.
- Un mejor mantenimiento de las aplicaciones proporciona un servicio con menos interrupciones, y que se resuelven en un menor espacio de tiempo.

Esta mejora se persigue mediante dos aproximaciones: Por un lado la implantación de una **herramienta para la gestión del ciclo de vida de las aplicaciones** y por otro lado la introducción de **metodologías ágiles de desarrollo y buenas prácticas**.



Buenas prácticas. Scrum:

Nuestro entorno de trabajo incluye más de 100 aplicaciones y 8 equipos de desarrollo, y los requisitos no siempre se conocen en los primeros estadios del proyecto. Todo esto hace que la aplicación las buenas prácticas recogidas por las metodologías ágiles se adapte mejor al escenario real que los clásicos ciclos de vida en cascada. En particular, en el caso descrito en esta comunicación, se está aplicando Scrum.

Herramientas. Team Foundation Server:

El área de desarrollo reparte su tiempo entre la gestión de proyectos y la resolución de incidencias derivadas de los proyectos anteriores.

Esta gestión de proyectos cubre al menos estos aspectos:

- Planificación de proyecto.
- Monitorización y control de proyecto.
- Gestión de acuerdos con el cliente final y con el proveedor.

Hasta ahora todas las tareas anteriores se hacían de forma manual sin el soporte de una herramienta. No es que el mero uso de la herramienta sea mejor en sí mismo pero sí es el instrumento que permite automatizar las tareas y en definitiva industrializar el proceso de generación de software.

La implantación de la herramienta de Team Foundation Server en la SGTIC tiene como objetivo pasar al nivel de madurez Gestionado, en la terminología de niveles de madurez de CMMI, donde los proyectos de la organización se planifican y se monitorizan, proporcionando al personal de la organización de los recursos adecuados para producir resultados controlados.

4. Introducción y objetivos. La calidad en el desarrollo software

¿Cuál es el mayor obstáculo para ofrecer servicios de calidad a los ciudadanos? ¿Por qué hay proyectos en los que se invierte mucho esfuerzo, tiempo y recursos, pero que no consiguen los resultados esperados?

La respuesta es sencilla y conocida por todos: por muchos recursos de los que se disponga, sin una buena gestión de ellos no se consigue nada.

Más difícil es saber cómo gestionar bien los recursos, y esa es precisamente la tarea diaria de todos.

En todas las organizaciones, en los últimos años, se han hecho esfuerzos para recoger y compartir las "mejores prácticas" en la gestión de proyectos. Los planes de calidad han sido impulsados también desde la normativa española, como desde el RD 951/2005, de 29 de julio, que establece el marco general para la mejora de la calidad en la AGE.

En el caso particular del desarrollo, CMMI sería el modelo de madurez más aplicado, y aunque no es el modelo aplicado en nuestro caso, describe una escala de niveles de madurez que sí serían aplicables.



Así, en el nivel de madurez 1 ("Inicial"), CMMI coloca aquellas organizaciones en las que los procesos son generalmente ad-hoc y caóticos. La organización generalmente no proporciona un entorno estable para dar soporte a los procesos. El éxito en estas organizaciones depende de la competencia y heroicidad del personal de la organización y no del uso de procesos probados. A pesar de este caos, las organizaciones de nivel de madurez 1 a menudo producen productos y servicios que funcionan; sin embargo, frecuentemente exceden sus presupuestos y no cumplen sus calendarios.

Las organizaciones de nivel de madurez 1 se caracterizan por una tendencia a comprometerse en exceso, a abandonar los procesos en tiempos de crisis y a una incapacidad para repetir sus éxitos.

El siguiente nivel sería el objetivo buscado por la iniciativa descrita en la presente comunicación: el nivel de madurez 2 ("Gestionado"). En el nivel de madurez 2, los proyectos de la organización han asegurado que los procesos se planifican y realizan de acuerdo a políticas; los proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados; involucran a las partes interesadas relevantes; se monitorizan, controlan y revisan; y se evalúan en cuanto a su adherencia a sus descripciones de proceso.

La disciplina de proceso reflejada por el nivel de madurez 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados. En el nivel de madurez 2, el estado de los productos de trabajo y la entrega de los servicios son visibles a la dirección en puntos definidos (p.ej., en los hitos principales y al finalizar las tareas principales). Se establecen compromisos entre las partes interesadas relevantes y se revisan, según sea necesario. Los productos de trabajo se controlan de forma apropiada. Los productos de trabajo y servicios satisfacen sus descripciones de proceso especificadas, estándares y procedimientos.

En la presente comunicación queremos mostrar una iniciativa concreta que se está realizando actualmente en la SGTIC del Ministerio de Economía y Hacienda en la dirección de mejorar la calidad del software desarrollado y mantenido, detallando tanto las acciones emprendidas, como los resultados buscados de mejora del servicio al ciudadano.

Es algo evidente, pero no siempre lo tenemos en cuenta: los servicios al ciudadano no mejoran solo cuando mejora el front-office, aquella parte de la Administración directamente visible al ciudadano. También logramos el objetivo de que los ciudadanos tengan un mejor servicio cuando trabajamos en la mejora del back-office, incluyendo esto desde los procesos que sustentan directamente los procedimientos administrativos, como también los sistemas de comunicaciones, infraestructuras, pasando también por aquellos servicios a los empleados públicos que les hacen su trabajo más productivo, y el ambiente laboral mejor.

Así, en la presente comunicación, presentamos un proyecto de mejora cuyos destinatarios directos, aparentemente, están lejos del servicio al ciudadano: se trata de equipos de desarrollo software, programadores, analistas, jefes de proyecto, jefes de servicio y de área de la SGTIC, desarrollando y manteniendo aplicaciones cuyos usuarios principales son empleados públicos. Por eso, además de explicar las acciones realizadas pondremos de relieve los beneficios que para el ciudadano supone el lograr que el trabajo dentro de un



ministerio se haga con mayor orden, mayor claridad en los requisitos, con claridad respecto a la forma de trabajar, sin pérdidas de información, aumentando el conocimiento dentro de la organización y apoyándose en herramientas de trabajo fiables.

La situación de partida de la SGTIC del Ministerio de Economía y Hacienda presenta retos similares a los de cualquier gran organización actual:

- Se gestionan más de 100 aplicaciones informáticas de uso interno.
- Para ello, se trabaja con la colaboración imprescindible de una gran proporción de personal externo. Los funcionarios tienen que asumir el papel de gestores de multitud de proyectos, procurando mantener el conocimiento técnico en la organización, pero sin poder realizar por sí mismos todo el trabajo.
- El personal externo está organizado en más de 8 equipos diferentes, sólo para desarrollo y mantenimiento software (incluyendo 3 equipos específicos sólo para mantenimiento). Estos equipos tienen que interactuar entre sí, y con muchos usuarios con requisitos dispares. La tarea de coordinación y gestión del trabajo realizado requiere mucho esfuerzo y tiempo, del que a menudo no se dispone. No se llega a todo lo que se quisiera y se plantea la posibilidad de evolucionar hacia formas de trabajo similares a las de una "software factory".

En esta situación, se plantea un proyecto que facilite y mejore la gestión que se realiza de todos los equipos y todas las aplicaciones software que se gestionan.

5. Implantación de Microsoft Team Foundation Server y plantilla de Scrum

5.1. Buenas prácticas en la gestión del ciclo de vida de las aplicaciones. Aspectos a cubrir

Cuando se parte de una situación compleja inicial, sobre todo a una carga de trabajo relativamente grande sobre las distintas personas involucradas y los distintos roles que muchos presentan, el primero de los pasos a dar es ir introduciendo distintas buenas prácticas en el ciclo de desarrollo, elementos que nos permitirán sin lugar a dudas realizar una mejor implantación de una metodología en pasos futuros.

En el caso del desarrollo de aplicaciones en la SGTIC, los distintos procesos de implantación de buenas prácticas estarán basados o usan de una manera u otra elementos presentes en Visual Studio Team System y/o Team Foundation Server, puesto que estos productos son la base de la iniciativa que se describe en esta comunicación.

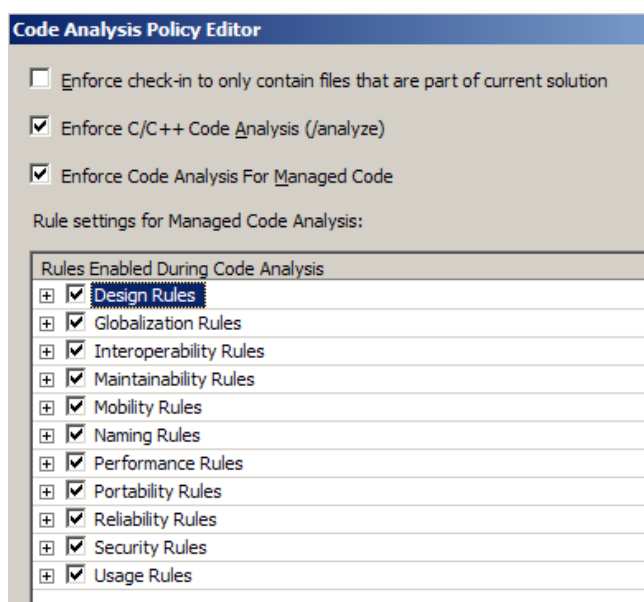
Así, los objetivos planteados en la presente iniciativa cubren los siguientes aspectos del ciclo de vida de las aplicaciones:

- **Modelado de aplicaciones y servicios:** El entorno de desarrollo es mayoritariamente .NET, y en la actualidad, se están usando las herramientas de modelado de Visual Studio 2003 y 2005. La implantación de Microsoft TFS incluye el despliegue de Visual Studio 2008, incluyendo herramientas de modelado más potentes. La migración de los proyectos de los entornos anteriores se hará a lo largo del año 2010, en un calendario fijado tomando en cuenta las prioridades del servicio y las necesidades de cada proyecto.

- **Análisis de código y modelado de clases:** TFS cuenta con un Analizador de código. Es una herramienta que permite mejorar el código, realizando recomendaciones antes o después de compilar. Puede configurarse bien para que sólo emita recomendaciones, bien para que no permita la integración del código desarrollado hasta que no se corrijan las deficiencias detectadas.

En este sentido, en la SGTIC ya existe previamente una listado de uso propio con todo un conjunto de condiciones de aceptación (NEMO). Además, como parte de la adopción de mejores prácticas, para todo el desarrollo en el entorno .NET se recomienda el seguimiento de las [guías de diseño de .NET framework](#) de Microsoft.

La herramienta TFS permite, por ejemplo, reglas como las mostradas en la siguiente imagen:



- Pruebas unitarias y cobertura de código: la herramienta implantada, TFS, permite la realización de pruebas de manera integrada con las el resto de pasos de desarrollo (compilación, subidas de código al repositorio, despliegues...). No es necesario recurrir a herramientas externas para realizar pruebas, pues el propio entorno suministra diversos tipos de proyectos que enfocan diversos tipos de pruebas. Se consigue así una mayor frecuencia en la realización de las pruebas unitarias.

Se irán adoptando dentro de la herramienta gradualmente y según las necesidades de cada proyecto los siguientes tipos de pruebas, con plantillas para implementarlos de que dispone la herramienta. Hay que desarrollarlos en cada caso, pero se almacenan y ejecutan desde TFS:

- Unitarios, que prueban métodos individuales de clases. Este tipo de pruebas obligan a mantener la calidad:
 - Facilita la puesta en práctica de buenas prácticas como las 'compilaciones diarias'
 - Permiten detectar tempranamente errores de rendimiento.
 - Facilitan la validación de requisitos



- Ayudan a entender y documentar las clases viendo los test unitarios y a que se destinan.
 - Permite observar la portabilidad del software al poder realizar las pruebas de forma sencilla en otras plataformas.
 - Ayudan a mejorar el soporte al usuario
 - Incrementan la modularidad y evitan el acoplamiento.
- Web Tests: Lanzan scripts de peticiones contra una aplicación web
 - Pruebas manuales: Listados de Instrucciones para realizar pruebas
 - Pruebas genéricas: Utilizan un programa externo para realizar un test
 - Pruebas de carga:
 - Crean un conjunto de pruebas a partir de diversos test
 - Coloca contadores en las máquinas que intervienen en el test para medir el rendimiento y otros aspectos
 - Pruebas ordenadas: Realización de cualquier combinación de las pruebas en un orden establecido.

Al estar integrado con la aplicación de control de código, nos proporciona así el dato de **cobertura de código**, que nos permite saber qué partes del código han sido probadas.

- **Compilación centralizada automática:** Basada en un lenguaje interpretado, MSBuild, permite adaptar el proceso de compilación para que automatice todo lo que sea posible, como despliegues, comprobaciones automáticas, incluso el envío de notificaciones de éxito o fracaso. Así la compilación se hará de forma integrada con el proceso de desarrollo, incluyendo la Ejecución de pruebas, el Análisis de código, versionado y la propia planificación temporal y de tareas del proyecto (mediante lo que se conoce en TFS como "elementos de trabajo")
- **Gestión de la configuración:** Por 'gestión de la configuración' se entienden el conjunto de procesos destinados a asegurar la validez de cualquier producto obtenido durante el ciclo de desarrollo mediante el control de los cambios realizados sobre los mismos y la disponibilidad de versiones estables de cada uno de los elementos. Esto incluye los aspectos de control de cambios, control de versiones y compilaciones, y aspectos de arquitectura como la definición de librerías de uso común disponibles para todos los proyectos en un repositorio único.

Control de cambios:

Actualmente, se está haciendo el control de código en la SGTIC con Microsoft SourceSafe, una herramienta que ha permitido que todo el código esté en un único repositorio, con control sobre qué versiones existen y cuáles están en producción. Pero esta herramienta se está utilizando de manera intensiva (con más de 100 proyectos de software, todos ellos con varias versiones históricas, y más de 60 usuarios). Se ha hecho por eso necesario pasar a una herramienta con mayor capacidad de trabajo, que además proporcionará todas las funcionalidades adicionales que tiene TFS.

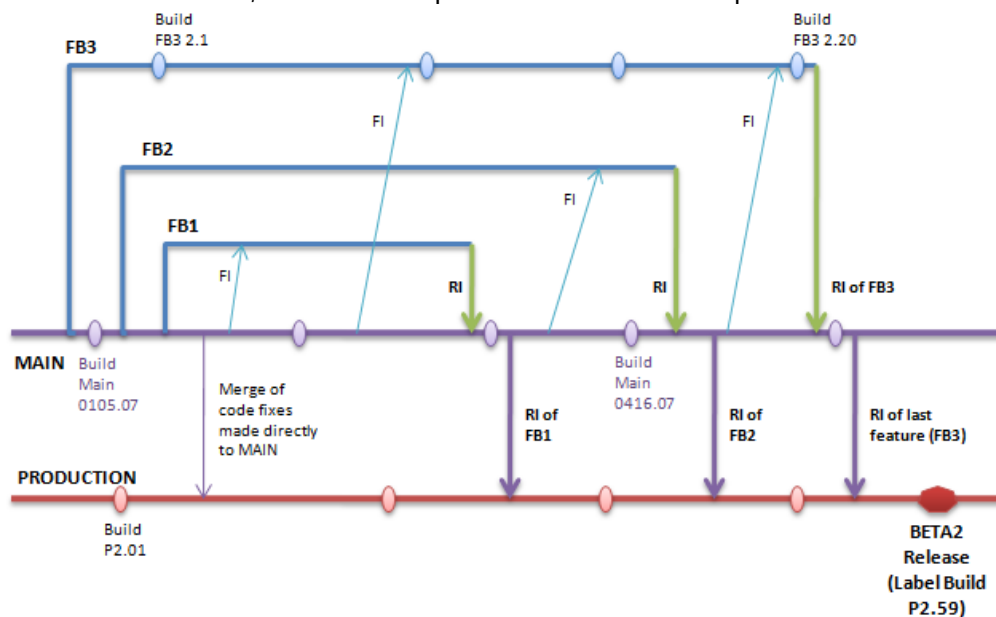
Como parte de las mejoras e implantación de buenas prácticas en el control de cambios está realizando en la SGTIC un manual de procesos que dicte las normas sobre el uso de la rama o ramas del control de código fuente de Team Foundation Server, estableciendo políticas de control de calidad antes de subir el código al repositorio centralizado común (Check In Policies). TFS permite establecer políticas de Check- In, referentes a pruebas, análisis de código, etiquetado del código, incluso a que el código esté asociado a un requisito que previamente haya sido validado por el jefe de proyecto del Ministerio de Economía y Hacienda.

Control de versiones:

En la actualidad MEH dispone de tres entornos dentro del ciclo de las aplicaciones, el entorno de desarrollo, preproducción y finalmente el entorno de producción. Muchas o algunas de las aplicaciones puestas en producción tienen una fase de mejora y soporte de deficiencias así como la creación de nuevas funcionalidades. Este escenario es lo que se conoce como un caso de 'Aislamiento por Mejora' dentro de los procesos de 'Bifurcaciones' o *branching*, esto es, el establecimiento de distintos entornos y por tanto distintos ámbitos de control del código.

Por ello, en la presente iniciativa se está haciendo una guía de *branching* que permita que todos los proyectos integrados dentro de TFS tengan un proceso común para dar soporte a aplicaciones existentes y la mejora de las mismas en distintas ramas. Este proceso común sin lugar a dudas beneficiará a los distintos miembros de los grupos de trabajo, puesto que tendrán documentado el proceso y podrán entender también con facilidad la configuración para el resto de proyectos del MEH.

En la siguiente imagen se puede ver un ejemplo de una guía de branching en Team Foundation Server, similar a la que se está diseñando para el MEH:



El control del repositorio de código es uno de los puntos que en un entorno con alta rotación de los desarrolladores, para poder asegurar que las versiones de los binarios puestos en producción se corresponden siempre con un código fuente localizable en caso necesario mediante el control de código.

Para esto se usará TFS, de forma que gestionará las versiones y garantizará que el código almacenado en el repositorio es el que se utiliza para la construcción de aplicaciones, y no el código depositado en la máquina de los desarrolladores.

Configuración de las soluciones:

Como parte de la implantación de TFS, se generarán también un conjunto de normas en cuanto a la configuración de las soluciones dentro de Visual Studio Team System.

Afectan por ejemplo a las normas de creación de proyectos, configuración de los diccionarios de lenguaje, información común de versionado e información de los



ensamblados, disposición de los ensamblados de terceros, o la situación de la documentación de los proyectos, escenarios de uso, manuales etc.

Visual Studio Team System y TFS pueden utilizarse para aplicar este conjunto de normas rápidamente: Así, se creará un proyecto con las librerías comunes, y cada proyecto contará con un portal de Sharepoint en el que se tendrá la documentación accesible vía web, con control de accesos y versiones.

5.2. Trabajo en equipo con TFS, y aplicación de una metodología de desarrollo Scrum.

La iniciativa que se describe en esta comunicación, además de las mejoras de herramienta y procedimiento en el ciclo de vida del software, ha planteado de forma simultánea la utilización de una metodología ágil, Scrum, aprovechando las características de la herramienta y el proceso de cambio iniciado.

Scrum ya se está utilizando con éxito desde hace casi un año en uno de los proyectos de desarrollo de la SGTIC, y en algunas de las empresas colaboradoras. Durante el año 2009 se realizó una sesión de formación, y como parte de la actual iniciativa se ha incluido una nueva sesión de formación, así como el uso de la plantilla metodológica "Scrum for Team System".

El objetivo es mejorar la gestión de los proyectos, y la forma de trabajo de los distintos equipos.

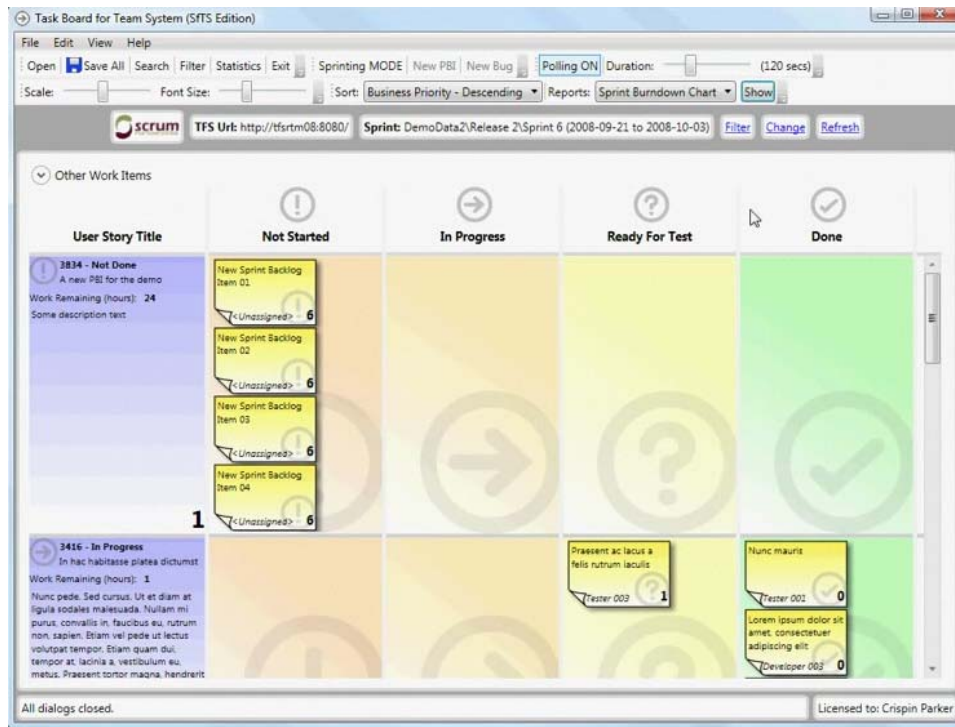
Team Foundation Server cuenta con soporte de metodologías: MSF, CMMI, personalizada, Scrum.

Esta última sería la metodología que se quiere impulsar para facilitar su uso con la herramienta, y para establecer los roles necesarios para su puesta en marcha.

Entre otras cosas, Scrum plantea un modelo de gestión de requerimientos que se corresponde con la estructura de múltiples equipos de desarrollo existente, pero que se querría mejorar. El "listado de requisitos" en Scrum se conoce como Product backlog. En él se incluyen historias de usuario, requisitos que deben suponer un aumento del valor para la organización (lo que en Scrum se resume con el acrónimo INVEST). Cada requisito debe ser:

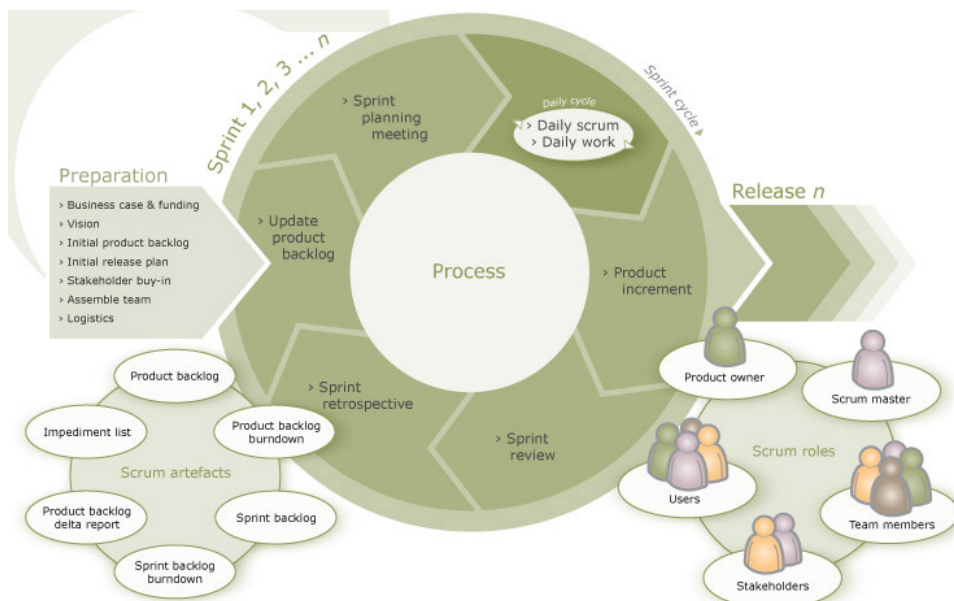
- **I**ndependent: no necesito otro requisito para tener que implementar este
- **N**egotiable: que todo el mundo pueda entender lo que se pone ahí, para poder opinar sobre ello.
- **V**aluable: que proporcione valor, algo que haga que el sistema sea mejor para el que lo usará
- **E**stimable: el esfuerzo que llevará hacerla
- **S**mall: que sea abordable por un equipo de proyecto de forma concreta, en un periodo de tiempo medible, lo que se llama en Scrum una iteración ó sprint, que suelen ser de una duración típica de un mes. Cuando una historia de usuario no es demasiado pequeña, se la llama más que "historia de usuario", una "épica". Lo ideal es que en cada iteración sea posible hacer varias (5 ó 6) historias de usuario
- **T**estable

Se muestra en la figura cómo se gestionarían los distintos requerimientos, convertidos en tareas para un equipo de trabajo:



Así, mediante esta plantilla de Scrum, se gestionará el Product Backlog de cada aplicación: esto es, el control de los requisitos y de su prioridad, estableciendo al final de iteraciones cortas demos en las que se muestra al usuario final los resultados obtenidos. Así, aumenta la calidad (que el usuario comprueba cada mes) y la implicación del equipo. La experiencia indica que aumenta mucho la motivación de los equipos.

Un esquema breve del ciclo de Scrum se muestra en la siguiente figura, tal y como se está implantando en los proyectos de esta iniciativa:



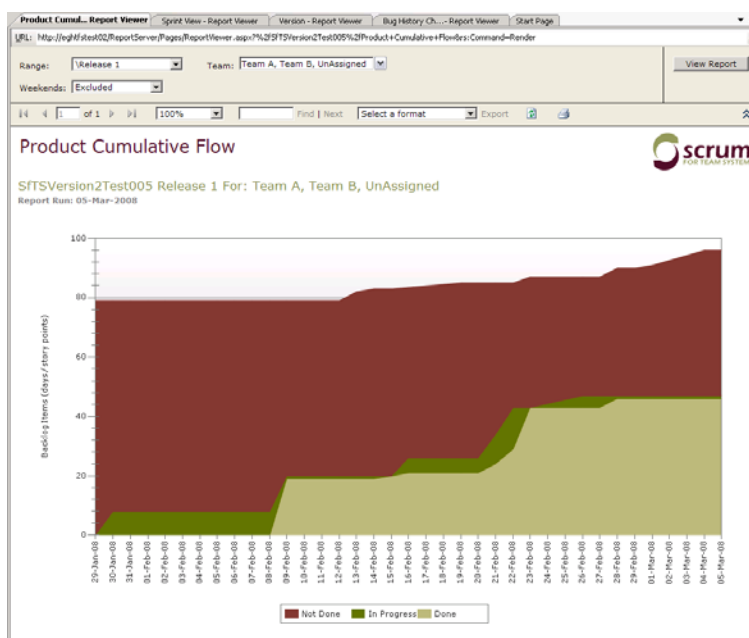
Mediante TFS se haría la gestión de elementos de trabajo: Asignación, documentación, trazabilidad, workflow, y será accesible desde múltiples herramientas:

- Visual Studio

- Microsoft Project
- Excel
- En caso necesario, integrado en otras herramientas por servicio web

La gestión de los requisitos y del trabajo realizado al desarrollar genera un conjunto de informes, de los que los más importantes son dos:

1. Sprint Cumulative Flow: El que muestra el trabajo que queda por hacer durante un mes, las tareas que ya están asignadas a un equipo. Permite detectar dentro del propio mes las desviaciones en el trabajo estimado, así como ir estableciendo, a medio plazo, cuál es la velocidad de trabajo de ese determinado equipo:
2. Product Cumulative Flow: El informe que muestra el trabajo realizado para una aplicación a lo largo de todo su ciclo de vida, permitiendo también detectar cuándo se producen aumentos en el trabajo esperado, para tenerlo en cuenta en los recursos necesarios o el plazo de entrega esperable. Muestra de manera clara cuánto trabajo queda por realizar en cada momento:



5.3. Conclusiones y planificación de la implantación de Team Foundation Server en la SGTIC del Ministerio de Economía y Hacienda.

El proyecto abordado supone una oportunidad de mejora interna en el desarrollo y mantenimiento de las aplicaciones del Ministerio de Economía y Hacienda.

Estas mejoras tienen su reflejo, indirecto, en los servicios que se prestan al ciudadano: menores costes de desarrollo, mayor calidad en las aplicaciones desarrolladas, menores tiempos de respuesta ante posibles incidencias y, en resumen, una mayor cercanía a las necesidades de los ciudadanos.

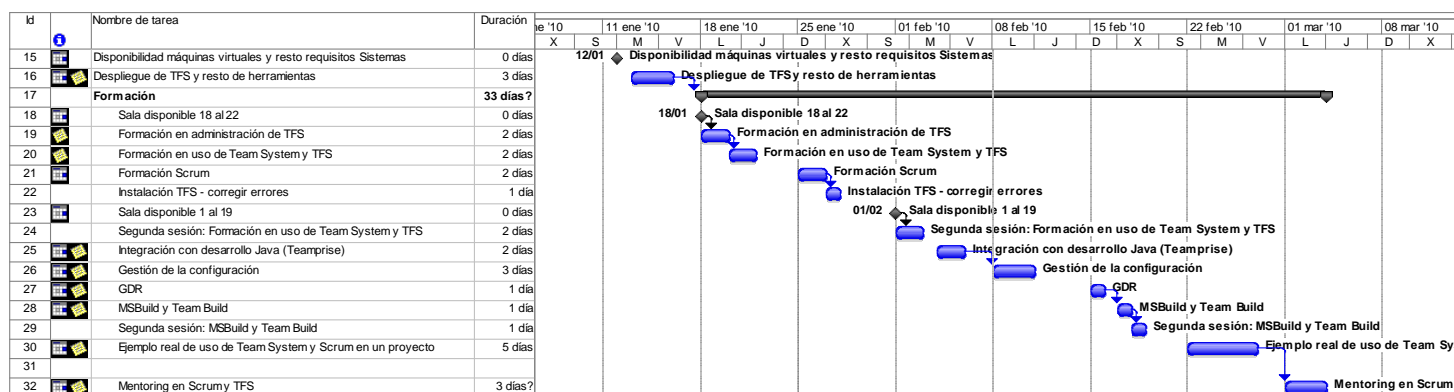
Todo lo anteriormente expuesto supone la introducción de cambios en los procedimientos y herramientas de las SGTIC del Ministerio de Economía y Hacienda. Aunque supongan mejoras, es necesario gestionar el cambio para que

llegue a toda la organización, incluyen los equipos de colaboradores externos, y para que se adopten gradualmente las mejoras que la organización considere como suyas.

Por esto, y partiendo de la experiencia de implantaciones similares en otras organizaciones, se ha realizado una planificación que incluye:

1. En primer lugar, la realización de un piloto, en febrero de 2009, en el que se estudió la viabilidad de la migración desde SourceSafe y las funcionalidades de TFS.
2. En enero de 2010, una etapa de formación. Esta formación se ha realizado tanto en entornos de prueba (máquinas virtuales en un aula de formación) como sobre el despliegue en producción de TFS. Se han impartido sesiones de formación entre uno y cinco días de:
 - Formación en administración de TFS
 - Formación en uso de Team System y TFS
 - Formación Scrum
 - GDR
 - MSBuild y Team Build
 - Integración con desarrollo Java (Teamprise)
3. Sesión de trabajo en equipo con todas las áreas para definir las políticas de Gestión de la configuración, y el calendario de migración de proyectos.
4. Ejemplo real de uso de Team System y Scrum en un proyecto: con un primer equipo de trabajo, una semana dedicada a migra los proyectos e implantar las primeras mejoras en la forma de trabajo, contando con la asistencia del formador en TFS y Scrum.

Se muestra a continuación la planificación del proyecto realizado:



5. Por último, se arrancará el plan de migración de todos los proyectos, con una duración estimada de seis meses, apoyado por un programa de mentoring en Scrum y TFS. La propia migración se aborda como un proyecto con metodología Scrum, definiendo un listado de requisitos (product backlog) y una serie de iteraciones en las que realizar todas las tareas de forma controlada.