

# Metodología Ágil para equipos pequeños usando plataformas Microsoft

Agile Methodology for Small Teams Using Microsoft Platforms

## Andrés Felipe Loboguerrero

Investigador – Grupo de Informática y Telecomunicaciones (i2t)  
Universidad Icesi, Cali (Colombia)  
andres.paz@correo.icesi.edu

## Lorena Castañeda Bueno

Profesora Asistente – Departamento de Tecnologías de Información y Comunicaciones  
Universidad Icesi, Cali (Colombia)  
lcastane@icesi.edu.co

## Hugo Fernando Arboleda, PhD.

Profesor Asociado – Departamento de Tecnologías de Información y Comunicaciones  
Universidad Icesi, Cali (Colombia)  
hfarboleda@icesi.edu.co

Fecha de recepción: Agosto 16 de 2011

Fecha de aceptación: Septiembre 12 de 2011

### Palabras clave

Equipos de desarrollo de software; metodologías de desarrollo de software; desarrollo ágil; gestión de proyectos; colaboración; Microsoft Team Foundation Server.

### Keywords

Software development teams; software development methodologies; agile development; project management; collaboration; Microsoft Team Foundation Server.

## Resumen

Con el objetivo de mejorar el rendimiento de los equipos de desarrollo de software se han definido muchas metodologías de desarrollo, entre las que se encuentran las prescriptivas y las ágiles. Pese a esto, los equipos no suelen emplear ninguna por no encontrarlas ajustadas a su contexto particular, y se van directamente a la tarea de programar. En este artículo se propone una metodología de desarrollo de software ligera y adaptada a un contexto preciso y bien definido, que alinea *Microsoft Solutions Framework for Agile Software Development (MSF4ASD)* con los lineamientos de gestión de proyectos presentados en la Guía del PMBOK. Además, se brindan elementos para la implementación de una plataforma Microsoft de desarrollo de software en equipo para soportarla.

## Abstract

Aiming to improve the performance of software development teams, many development methodologies have been defined; among which are the prescriptive and agile. Despite this, teams do not usually use any of these by not finding them tailored to their particular context, and instead go directly to the task of programming. This article proposes a light software development methodology adapted to a precise and well-defined context, aligning *Microsoft Solutions Framework for Agile Software Development (MSF4ASD)* with the project management guidelines presented in the PMBOK Guide. In addition, it offers elements for the implementation of a Microsoft team software development platform to support it.

---

## I. Introducción

Un equipo de desarrollo de software está compuesto por personas que realizan funciones diferentes, y cada una de ellas puede ser responsable de una o más de estas funciones. Por ejemplo, un integrante puede estar concentrado en la gestión del proyecto mientras que otros pueden participar tanto en el diseño del sistema como en su implementación. Sin embargo, hay varios aspectos que entran a restringir el buen desarrollo de estas funciones. Por un lado, están las condiciones del equipo —es decir, sus procesos o metodologías—, las habilidades de cada integrante, los roles que éstos desempeñan y su disponibilidad (Humphrey, 2005; Sawyer, 2004). Por otro lado, están las condiciones del proyecto, como fechas, requerimientos y el mismo contacto con el cliente (Phillips, 2004). Finalmente, están las condiciones de las herramientas con las que cuenta el equipo. Varias de estas condiciones intrínsecas y extrínsecas de los equipos no se pueden tratar o cambiar, por lo que es necesario enfocarse en los aspectos metodológicos y técnicos.

En el campo del desarrollo de software se han definido diferentes metodologías; entre ellas están las prescriptivas y las ágiles. De las prescriptivas puede ser un buen representante el proceso TSP (Humphrey, Chick, Nichols, & Pomeroy-Huff, 2010); estas metodologías valoran la documentación y la planificación disciplinada. Las ágiles (Agile Alliance, 2001), entre las que se encuentra Extreme Programming (XP) (Beck, 2004), se guían por la interacción entre las personas, la respuesta al cambio y el software funcional. Cada una de esas metodologías tiene aspectos positivos y negativos, y es el equipo quien debe decidir, de acuerdo con sus características, cuál emplear para un determinado proyecto. Por ejemplo, los desarrolladores de software que trabajan con tecnologías Microsoft tienen otro conjunto de posibilidades; cuentan con el Microsoft Solutions Framework (MSF) para CMMI (MSF4CMMI) y MSF para Desarrollo Ágil de Software (MSF4ASD). Todos estos marcos metodológicos necesitan adaptarse de acuerdo con los contextos precisos y bien definidos en los que se encuentran los equipos o empresas que los utilizan.

Otro desafío al que se enfrenta el equipo es lograr que todos sus integrantes colaboren sin problemas. Tratar de integrar lo que cada miembro del equipo realiza individualmente no es una tarea fácil y requiere de mucha coordinación. David, Gousset y Gunvaldson (2006) afirman que por lo general cada función en el equipo se apoya en el uso de alguna herramienta específica. Por ejemplo, un desarrollador trabaja con un IDE según la naturaleza del proyecto y no tiene ninguna dificultad en tratar temas de desarrollo. Los arquitectos de software elaboran el diseño de la aplicación apoyándose en alguna herramienta de modelamiento. Algo similar sucede con los encargados de las pruebas que utilizan sus herramientas de prueba. Los directores de proyecto, por el contrario, especifican casos de uso y escenarios en hojas electrónicas y administran las etapas del

proyecto con un programa de gestión de proyectos. Es también responsabilidad del director del proyecto mediar entre los miembros del equipo y establecer reuniones para actualizar el estado del proyecto y rastrear las entregas de cada uno. Como se observa, el grupo puede llegar a tener herramientas de trabajo de diversa naturaleza.

Los equipos de desarrollo de software requieren de un marco metodológico de desarrollo que además de estar adaptado a un contexto específico tenga en cuenta los aspectos técnicos para ser una guía efectiva en el ciclo de vida de sus proyectos. De igual forma, necesitan contar con una plataforma de desarrollo que les provea un ambiente real de colaboración durante el avance de sus asignaciones. En este artículo presentamos una metodología ligera para desarrollo de aplicaciones en un contexto bien identificado en que los equipos usan, como la base de su desarrollo, herramientas Microsoft, desarrollan sus proyectos de forma iterativa e incremental, tienen ciclos de desarrollo cortos y se componen aproximadamente de cinco personas.

Nuestra principal contribución, además de identificar y describir un contexto de desarrollo de software para el cual aplica esta propuesta, está en que alineamos un marco metodológico de desarrollo de software conocido y aceptado, MSF4ASD, con los lineamientos de la gestión de proyectos que se presentan en el Project Management Body of Knowledge (PMBOK).

El resto de este artículo se organiza de la siguiente manera: la sección II da el contexto de estudio y describe las características de los equipos de desarrollo. En la sección III se describe el marco metodológico MSF4ASD. La sección IV habla de los lineamientos de gestión de proyectos del PMBOK. La sección V presenta la metodología propuesta junto con las herramientas de soporte. Y finalmente, en la sección VI se exponen las conclusiones de la presente investigación y se referencian algunos trabajos futuros.

## **II. Contexto de estudio**

Los equipos de desarrollo de software se pueden encontrar en muchos contextos. En un extremo están los equipos de una sola persona, es decir, desarrolladores que trabajan solos. Estos equipos suelen trabajar en proyectos completamente informales, como por ejemplo los esfuerzos de desarrollo de software open source, donde no hay unos objetivos, un alcance, un presupuesto o una fecha de finalización definidos explícitamente. En el otro extremo están los grandes equipos de desarrollo, con proyectos que tienen largos ciclos de vida y requieren de procesos altamente estandarizados y de un registro de todas las decisiones tomadas. La metodología que aquí se propone está diseñada para equipos en un contexto muy preciso. Los equipos deben ser pequeños, de alrededor de cinco personas, cuyos integrantes no pueden reunirse con frecuencia debido a sus múltiples responsabilidades externas al proyecto de desarrollo. Los proyectos deben reunir las siguientes características: su tamaño debe ser relativamente pequeño con respecto a la funcionalidad implementada; deben ser informales, es decir, no es necesario

probar el cumplimiento de procesos estandarizados; deben desarrollarse de forma iterativa e incremental en ciclos cortos, de una duración no mayor a dos semanas, sobre un cronograma de entrega rápido; y su base del desarrollo debe estar en herramientas y tecnologías Microsoft.

Muchos de estos equipos de desarrollo de software que cuentan con tan poco recurso humano no siguen una serie de pasos planeados y estructurados para realizar sus proyectos. Cada proyecto realizado le deja al equipo desarrollador experiencia y aprendizaje muy valiosos, pero no suele quedar un registro de las lecciones aprendidas, buenas y malas, que ayuden a mejorar sus procesos para futuros proyectos (Pino, García, Piattini, & Oktaba, 2006). Es posible que cuando el equipo logre incorporar nuevas prácticas en sus procesos lo haga de manera poco organizada. Esta realidad hace de la tarea de desarrollar el proyecto una experiencia traumática que repercutirá, entre otros, en la calidad del producto presentado. Por calidad se entiende el cumplimiento con las expectativas del grupo y del cliente frente a la solución, del presupuesto y en el tiempo de entrega. Incumplir en cualquiera de estos tres aspectos tendrá un efecto negativo en el grupo del proyecto, que podrá perder la oportunidad de llegar a desarrollar futuros proyectos.

Uno de los grandes desafíos que siempre se presenta al momento de trabajar en un equipo de desarrollo de software –y que disminuye su rendimiento– es lograr que todos sus miembros colaboren sin problemas. Si el grupo de trabajo desea ser productivo en el desarrollo de software, debe tener claro que debe contar con un ambiente de colaboración para que las personas trabajen en equipo. Es importante entender que no importa el rol que se desempeñe en el proyecto; si se incumple con la fecha de entrega o con los estándares de calidad especificados todo el equipo de trabajo debe asumir la responsabilidad en el fracaso. Por esta razón es importante tener un intercambio desinteresado de información, puesto que si cada uno de los integrantes conoce la información de sus demás compañeros se eliminan los retrasos generados al emplear mucho tiempo en reuniones de actualización del estado del proyecto y en rastrear las entregas de todos, así como la integración y redundancia de desarrollos.

### **III. Marco metodológico MSF4ASD**

MSF4ASD incorpora ideas claves del movimiento Ágil junto con los principios y prácticas definidos en MSF (Mak, 2006; Sutherland, 2010). El marco metodológico representa una versión de MSF que le resta importancia al gobierno formal de los proyectos para hacer énfasis en los principios ágiles. Este marco metodológico toma como base principal a Scrum (Schwaber, 2004; Schwaber & Beedle, 2001) para ejecutar los proyectos siguiendo una estrategia de desarrollo iterativa, y un enfoque basado en escenarios y en el contexto de realización del proyecto. Lo anterior le permite tener procesos adaptables y la habilidad para hacerle frente

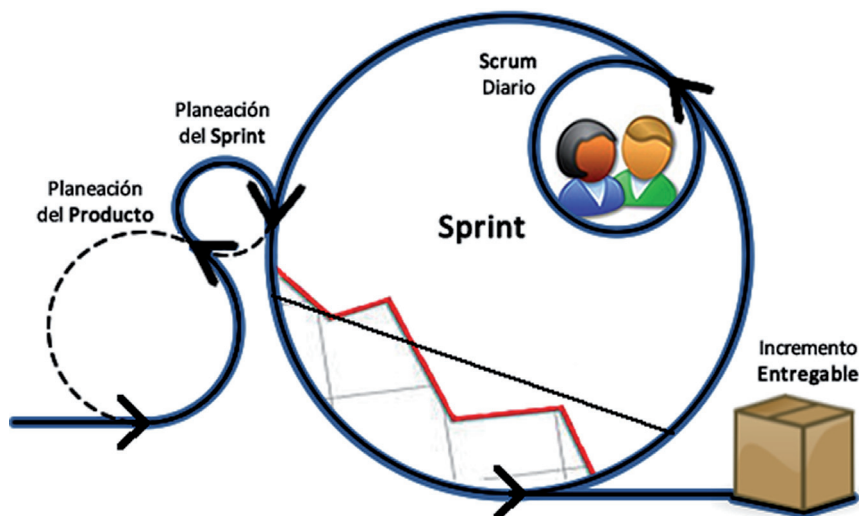


Figura 1. Procesos de MSF4ASD, adaptada de Microsoft (Microsoft, 2010a)

a los ambientes cambiantes (Traa, 2006). MSF4ASD está dirigida a equipos de desarrollo orientados a resultados que soportan la construcción de aplicaciones orientadas a objetos .NET, Web y Web Service, entre otras aplicaciones orientadas a objetos (Mak, 2006), y que sean capaces de trabajar con muy poca documentación –y que además se encuentre incompleta–.

En este marco metodológico se define un conjunto de procesos y actividades que le ayudan al equipo de desarrollo a entregar un producto de mayor valor para el cliente en menos tiempo. Estos procesos y actividades también les permiten a los clientes una interacción mayor con el equipo durante el tiempo de desarrollo. En la Figura 1 se resume la interacción entre estos procesos. Lo primero que el equipo debe hacer es crear un plan de alto nivel para el proyecto, que se irá actualizando a lo largo de su ejecución.

MSF4ASD no trata de definir todo el producto al inicio del proyecto, sino que el equipo realiza iteraciones cortas, o *sprints*, para ir refinándolo. Cada *sprint* inicia con una planeación en la cual se establece su alcance. Durante el *sprint* el equipo completa las tareas de implementación, hace seguimiento diario del progreso (*daily scrum*) y se asegura de cumplir los criterios de aceptación del cliente. Al finalizar un *sprint* se hace una revisión global del progreso del proyecto y se procede a planear la siguiente iteración. Cuando ya se han completado todos los *sprints* se hace el *release* del producto y se mira en retrospectiva todo lo que sucedió.

## IV. Lineamientos de gestión de proyectos del PMBOK

La Guía del PMBOK se presenta como un estándar reconocido para la gestión de proyectos (Project Management Institute, 2008). Aquí se identifica un conjunto de procesos de dirección de proyectos que se agrupan en cinco categorías, mejor conocidas como grupos de procesos. Estos son:

- **Grupo de procesos de iniciación.** Aquellos que se realizan para definir ya sea un nuevo proyecto o una nueva fase de uno existente. Una fase es una división del mismo proyecto que requiere de un componente de gestión específico. El conjunto de todas sus fases constituye el ciclo de vida del proyecto.
- **Grupo de procesos de planificación.** Aquellos que establecen el alcance, los objetivos y el curso de acción del proyecto.
- **Grupo de procesos de ejecución.** Aquellos que completan el trabajo definido en la planificación.
- **Grupo de procesos de seguimiento y control.** Aquellos que proveen el seguimiento y la regulación al progreso y desempeño del proyecto.
- **Grupo de procesos de cierre.** Aquellos que finalizan formalmente el proyecto o una fase de este.

Los cinco grupos de procesos para la dirección de proyectos son necesarios en todo proyecto. Estos procesos se muestran como elementos diferenciados y con interfaces bien definidas en la Figura 2. En ese diagrama de flujo de procesos se proporciona un resumen global del flujo y las interacciones básicas entre los grupos de proceso. Sin embargo, estos pueden superponerse o interactuar en formas totalmente diferentes. Los grupos de proceso se relacionan entre sí mediante los resultados que producen. De esta forma, la salida de uno de estos procesos puede convertirse en la entrada de otro.

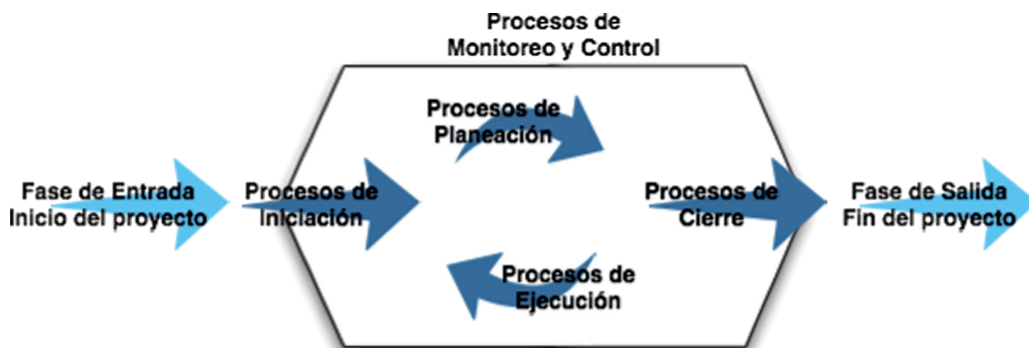


Figura 2. Grupos de procesos de la dirección de proyectos, tomada de Project Management Institute (Project Management Institute, 2008)

## V. Metodología propuesta y herramientas de soporte

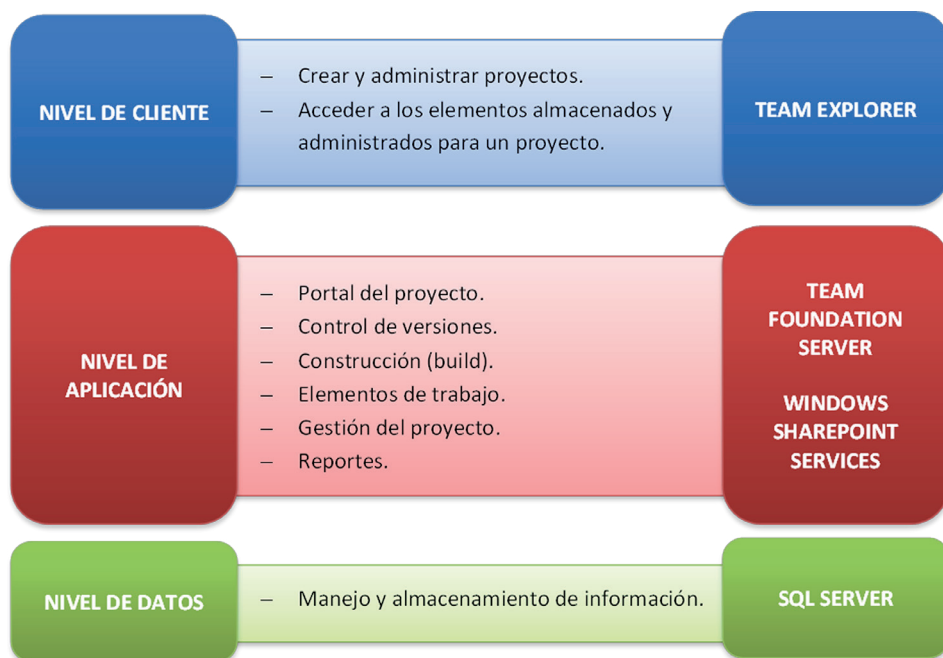
La metodología propuesta toma como base las prácticas y los procesos que se presentan en las secciones III y IV y los adapta al contexto presentado en la sección II. Sigue la agrupación de procesos planteada en la Guía del PMBOK, pero asocia el desarrollo ágil del producto de MSF4ASD. Al igual que en MSF4ASD, los procesos de la metodología propuesta no buscan especificar todo el producto al inicio del proyecto, sino lograr hacerlo a través de iteraciones en las cuales se desarrolle poca funcionalidad en un lapso muy corto de tiempo, de no mayor a dos semanas. El objetivo con cada una de estas iteraciones es obtener un demo funcional del software sin errores.

Antes de iniciar un proyecto se debe formar un equipo –si los integrantes cambian de un proyecto a otro– a través de cinco grupos de procesos similares a los establecidos en el PMBOK y nombrados en la sección anterior. Estos grupos son: iniciación, planeación, ejecución, seguimiento y control, y cierre. Estos grupos de procesos se componen a su vez de varios procesos que se detallan en las siguientes subsecciones. Las iteraciones incluyen los siguientes cinco subprocesos: diseñar, planear, codificar, probar y evaluar. Estos subprocesos se detallan en el grupo de proceso de ejecución. La caracterización de cada proceso, así como en el PMBOK, se va a componer de una breve descripción, sus respectivas entradas y salidas, y un conjunto de herramientas o técnicas para su correcta implementación.

En la Figura 3 se muestra el flujo global propuesto. Debido a que un proyecto es un esfuerzo temporal (Microsoft, 2010b), el grupo de procesos de iniciación lo comienza y el grupo de procesos de cierre lo termina. El grupo de procesos de seguimiento y



Figura 3. Grupos de proceso de la metodología propuesta



**Figura 4.** Arquitectura de la plataforma de desarrollo de software en equipo, adaptada de David et al. (2006) y Microsoft (2010b)

control interactúa con los demás grupos de procesos para poder ejercer el seguimiento de las actividades del proyecto. Conforme el proyecto avanza se realizan los procesos de los grupos de procesos de planeación y ejecución. Una iteración comienza con el subproceso de diseñar y termina con el subproceso de evaluar. A medida que avanza la construcción del producto se realizan las actividades de codificar y probar.

En la práctica estos grupos de proceso podrán superponerse e interactuar en formas que no se establecen en este documento.

La plataforma que Microsoft propone para el desarrollo de software en equipo provee herramientas dirigidas a apoyar y simplificar la colaboración entre los integrantes del equipo de desarrollo de software (Microsoft, 2010c). Esta va a permitir la comunicación con cualquier miembro del equipo y acceder a un conjunto de servicios, tales como un portal para el proyecto, control de versiones, motor de construcción, reportes y herramientas de comunicación. De esta forma se requieren menos reuniones de revisión y actualización del estado del proyecto y se tendrá más claridad con respecto al trabajo que se esté llevando a cabo. En la Figura 4 se presentan los tres niveles de la plataforma tecnológica de implementación: nivel de cliente, con Microsoft Team Explorer (TE; un complemento para MS Project y MS





**Figura 5.** Despliegue propuesto para la plataforma de desarrollo de software en equipo, adaptada de Microsoft (2010b)

Visual Studio); nivel de aplicación, con MS Visual Studio Team Foundation Server (TFS; servidor de colaboración) y MS Windows SharePoint Services (WSS; servicios de colaboración web para TFS); y nivel de datos, con Microsoft SQL Server (gestor de bases de datos).

La plataforma puede ser desplegada de varias formas, unas más complejas que otras. El despliegue que mejor se ajusta al contexto presentado en la sección II es el más simple de la plataforma, que, como se muestra en la Figura 5, emplea un solo servidor para manejar los niveles de aplicación y de datos e instala el nivel de cliente en cada uno de los equipos de los usuarios, quienes se conectan al servidor.

Microsoft ha introducido el concepto de plantilla de proceso para abordar la integración entre las metodologías de desarrollo y el soporte tecnológico. La plantilla de proceso define varios aspectos importantes de cada proyecto que afectan la forma como el equipo trabaja. Se pueden definir tipos de elementos de trabajo, consultas de elementos de trabajo, reportes de seguimiento, iteraciones, la seguridad del proyecto, las plantillas del portal del proyecto, notas de *check-in* para el control de versiones, entre otras opciones. Para esta propuesta se tomó como base la plantilla de MSF4ASD, que fue modificada para incluir las especificaciones de la metodología planteada.

Las siguientes subsecciones presentan un vistazo general de cada grupo de proceso de la propuesta, así como de cada proceso que lo conforma y el soporte tecnológico que se ofrece.

### **A. Grupo de procesos de iniciación**

Los procesos de iniciación comienzan el proyecto y se componen de aquellas actividades que definen el equipo de trabajo y el nuevo proyecto que será realizado. Como soporte

tecnológico el equipo cuenta con la biblioteca de documentos compartidos almacenada en TFS, a la que se puede acceder a través de TE. La biblioteca de documentos guarda todos aquellos documentos que serán compartidos entre los integrantes del equipo durante la realización del proyecto. Incluye solo un proceso:

- **Desarrollar el documento del anteproyecto.** Desarrollar el documento que presenta formalmente el proyecto, indicando los integrantes del equipo, los objetivos y las metas del proyecto y la descripción del proyecto, que incluye el propósito funcional, las características funcionales generales, los usuarios potenciales, el modelo de operación y la inversión requerida. El documento del anteproyecto se guarda en la biblioteca de documentos como documento de MS Word.

### ***B. Grupo de procesos de planeación***

El equipo dedicará la mayor cantidad de su tiempo a desarrollar el producto en una serie de iteraciones. Sin embargo, es necesario crear el plan de trabajo. Los procesos de planeación buscan definir el alcance y especificar tareas, entregables, recursos y tiempo necesarios para llevarlo a cabo. Aquí también interviene la biblioteca de documentos compartidos junto con MS Project como herramienta de planificación. Incluye los siguientes procesos:

- **Recopilar los requisitos y requerimientos del producto.** Definir y documentar los requisitos y requerimientos de alto nivel del producto, que incluyen las funcionalidades necesarias y deseables del producto y, opcionalmente, aspectos técnicos de éste (e.g. tiempo de respuesta, impacto en memoria o en disco, confiabilidad, tolerancia a fallos, seguridad). Aquí es necesario acudir a expertos en el tema del proyecto o usuarios finales que puedan apoyar estas actividades. El documento de requisitos y requerimientos del producto se guarda en la biblioteca de documentos como documento de MS Word.
- **Definir el alcance.** Desarrollar una descripción del alcance del proyecto a partir de los requerimientos definidos para el software, teniendo en cuenta restricciones, riesgos y supuestos. El documento del alcance del proyecto se guarda en la biblioteca de documentos como documento de MS Word.
- **Crear la estructura de desglose de trabajo (EDT).** Subdividir jerárquicamente el trabajo del proyecto en componentes entregables pequeños y fáciles de manejar. Con este proceso se inicia la creación del cronograma del proyecto, que se apoya en el uso de MS Project.
- **Definir las actividades.** Identificar y definir las actividades necesarias para elaborar los entregables. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.
- **Secuenciar las actividades.** Identificar y definir las interrelaciones entre las actividades. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.

- **Identificar responsabilidades.** Consiste en establecer y documentar las responsabilidades que tendrá cada uno de los integrantes del equipo. A cada miembro se le asignarán varias responsabilidades según sus habilidades pero manteniendo las cargas de trabajo balanceadas. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.
- **Asignar los recursos de las actividades.** Asignar las personas requeridas para ejecutar cada actividad. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.
- **Estimar la duración de las actividades.** Establecer la cantidad de tiempo necesario para finalizar cada actividad de acuerdo con los recursos asignados. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.
- **Desarrollar el cronograma.** Consolidar toda la información de las iteraciones y las actividades, sus interrelaciones, los recursos asignados a cada una y su respectiva duración, para crear el cronograma del proyecto. Se revisa todo el contenido del cronograma del proyecto desarrollado en MS Project.
- **Planificar las comunicaciones.** Determinar las necesidades de información entre los miembros del equipo (e.g. qué información se necesita; quién la necesita; quién la entrega, cómo y cuándo) y definir el cronograma de reuniones. Las reuniones deben tener un propósito específico y ocurrir a una frecuencia definida. Se sugiere que haya reuniones de evaluación para cada iteración en las cuales se revise el desempeño del equipo, el trabajo realizado y no realizado, se registren las lecciones aprendidas y se planteen propuestas de mejoramiento tanto para el producto como para el proceso. También se sugiere una reunión de cierre del proyecto para darlo por terminado, que debe registrar su desempeño global, revisar el trabajo realizado y no realizado, registrar lecciones aprendidas y plantear propuestas de mejoramiento del proceso. Las demás reuniones, como por ejemplo las de planeación o seguimiento, se realizan a criterio del equipo. Se agrega esta información al cronograma del proyecto desarrollado en MS Project.

### **C. Grupo de procesos de ejecución**

Los procesos de ejecución se componen de aquellas actividades que completan el trabajo que se ha definido en el cronograma del proyecto. Aquí estarán involucradas todas las herramientas de soporte. Incluye los siguientes procesos:

- **Desarrollo de las actividades.** Ejecutar el trabajo definido en el cronograma. Todo el trabajo definido se va a convertir en elementos de trabajo en TFS. Un elemento de trabajo es una sola unidad de trabajo que se necesita completar, es decir, una tarea necesaria para finalizar el proyecto. A medida que se van completando los elementos de trabajo el proyecto avanza. Un elemento de trabajo registra la descripción, la asignación, la prioridad y el estado del trabajo a realizar. Los elementos de trabajo se

crean usando MS Visual Studio. El producto de software se desarrolla de acuerdo con las iteraciones establecidas. Una iteración toma por lo general una semana, máximo dos, y produce un incremento funcional del software. Los subprocesos que componen una iteración son:

- **Diseñar.** Detalla y documenta los requerimientos del sistema que se desarrollarán en la iteración. La documentación se hace en la forma de historias de usuario (*user stories*) y un documento de arquitectura. Las historias de usuario son un tipo de elemento de trabajo que definen una característica del software a desarrollar. En ellas se especifica quién es el usuario, qué necesita hacer y por qué necesita hacerlo. La funcionalidad que se detalle deberá ser suficientemente pequeña para implementarse en una iteración; de lo contrario, deberá dividirse en historias más pequeñas. El documento de arquitectura incluye el modelo conceptual de la aplicación (e.g. componentes funcionales, arquitectura de la infraestructura, elementos de integración con otras aplicaciones o plataformas, flujo del proceso, actores involucrados). Este documento se guarda en la biblioteca de documentos como documento de MS Word.
- **Planear.** Establece el plan de trabajo para la iteración a partir de las historias de usuario previstas. Se especificarán tareas y otro tipo de elementos de trabajo que se crean para dividir una historia de usuario en actividades de desarrollo (implementación) y de pruebas. Una tarea representa una cantidad de trabajo que puede hacerse en entre una y dos horas por una persona.
- **Codificar.** Lleva a cabo las tareas de desarrollo de la aplicación siguiendo el plan de trabajo establecido y basándose en la especificación de las historias de usuario para la iteración y el documento de arquitectura.
- **Probar.** Realiza las tareas de prueba necesarias para descubrir errores en el código generado. En el caso de encontrar un error, se crea un elemento de trabajo de tipo error, que sirve para reportar los defectos encontrados.
- **Evaluar.** Finaliza una iteración revisando su progreso. También se revisan los errores a corregir o cambios a implementar en la siguiente iteración.
- **Distribuir la información.** Hacer disponible la información relevante de los entregables del proyecto para que esté a disposición de todo el equipo de trabajo. Se puede hacer uso del portal del proyecto. El portal del proyecto es el lugar para obtener información acerca del proyecto.

#### **D. Grupo de procesos de seguimiento y control**

Los procesos de seguimiento y control se componen de aquellas actividades que controlan el progreso y el desempeño del proyecto. Se pueden identificar partes del proyecto que requieran ajustes de fondo y forma; así se inician los cambios necesarios. Es necesario usar MS Visual Studio para acceder a los elementos de trabajo y MS

Project para realizar los ajustes al cronograma del proyecto. Incluye los siguientes procesos:

- **Verificar entregables del proyecto.** Revisar los entregables del proyecto que se han completado, ya sea para verificar su completitud, recomendar cambios o para aceptarlos. El proceso se ejecuta cada vez que sea necesario validar un entregable del proyecto.
- **Controlar el cronograma del proyecto o el plan de trabajo de la iteración.** Hacer un seguimiento al estado del proyecto o la iteración con el fin de actualizar su avance en el cronograma del proyecto o en el plan de trabajo de la iteración, respectivamente, y realizar ajustes, en caso de ser necesarios.

### ***E. Grupo de procesos de cierre***

Los procesos de cierre finalizan el proyecto; por lo tanto, se componen de aquellas actividades que concluyen las actividades de los demás grupos de proceso, para terminar formalmente el proyecto. Aquí estarán involucradas todas las herramientas de soporte. Incluye el siguiente proceso:

- **Cerrar el proyecto.** Finalizar todas las actividades de todos los demás grupos de proceso para completar el proyecto y documentar las razones por las que se dio por terminado antes de su culminación. La finalización de los elementos de trabajo se realiza a través de MS Visual Studio. Los documentos de lecciones aprendidas del proyecto y cierre del proyecto se guardan en la biblioteca de documentos como documentos de MS Word.

En la Tabla 1 se presenta un resumen de todos los procesos de la metodología propuesta con su respectivo soporte tecnológico.

## ***Conclusiones***

Los equipos de desarrollo de software que cuentan con muy pocos recursos humanos se enfrentan a varios desafíos que condicionan el buen desarrollo de sus funciones, debido a que no siguen un proceso planeado y estructurado para llevar a cabo sus proyectos. El entorno de desarrollo y el marco de trabajo que estos equipos establezcan entran a jugar un papel importante durante el transcurso de los proyectos. Un valor del equipo procede de una administración eficaz de los procesos e integrantes, lo que conlleva el aumento de la productividad a la hora de satisfacer los plazos y presupuestos de los proyectos, y la generación de software de alta calidad.

En tanto esta propuesta recoge las prácticas, procesos y otros elementos de MSF4ASD y de la Guía del PMBOK, se hace alineándolas hacia los

Grupo de Proceso	Proceso	Soporte tecnológico					
		Team Explorer		MS Word	MS Project	MS Visual Studio	Portal del proyecto en MS SharePoint
		Biblioteca de documentos	Elementos de trabajo				
<b>Iniciación</b>	Desarrollar el documento del anteproyecto.	X		X			
	Recopilar los requisitos y requerimientos del producto.	X		X			
	Definir el alcance.	X		X			
	Crear la estructura de desglose de trabajo (EDT).	X			X		
<b>Planeación</b>	Definir las actividades.	X			X		
	Secuenciar las actividades.	X			X		
	Identificar responsabilidades.	X			X		
	Asignar los recursos de las actividades.	X			X		
	Estimar la duración de las actividades.	X			X		
	Desarrollar el cronograma.	X			X		
<b>Ejecución</b>	Planificar las comunicaciones.	X			X		
	Desarrollo de las actividades.	X	X	X	X	X	
	Distribuir la información.	X	X		X		X
<b>Seguimiento y Control</b>	Verificar entregables del proyecto.	X	X		X		X
	Controlar el cronograma del proyecto o el plan de trabajo de la iteración.	X			X		
<b>Cierre</b>	Cerrar el proyecto.	X		X	X	X	X

Tabla 1. Procesos de la metodología propuesta con su respectivo soporte tecnológico

equipos pequeños que emplean herramientas Microsoft como la base de sus desarrollos. Esto con el fin de mejorar el rendimiento de estos equipos, apoyar la colaboración entre sus miembros y hacer los ciclos de vida de sus desarrollos más organizados. Sin embargo, la metodología planteada no ha sido puesta a prueba, por lo que este es un trabajo futuro que deberá realizarse. Así mismo, puede generar una resistencia en los equipos que la adopten, pues se deberán seguir los procesos propuestos y aprender a usar la plataforma. Para mitigar esta situación es necesario incluir un proceso de *mentoring* que facilite la adherencia de las personas a los procesos.

Esta metodología desarrollada se enfoca en un contexto específico, por lo que el problema aún seguirá presente en otros contextos donde las condiciones sean diferentes. Sin embargo, esta metodología puede servir como una base extensible que puede ser adaptada y mejorada para que sea útil ante nuevas condiciones.

La plataforma Microsoft de desarrollo de software en equipo es una herramienta muy valiosa, pero poco utilizada debido sus diferentes requerimientos técnicos y de capacitación. La ventaja principal que resalta esta propuesta es la integración con los procesos para que las personas puedan trabajar de manera más rápida y eficaz. Además, proporciona a los líderes de los proyectos una mayor predictibilidad y visibilidad en el ciclo de vida, lo que puede mejorar la gestión de los proyectos. **S&T**

## Referencias bibliográficas

- Agile Alliance. (2001). *Manifiesto for agile software development*. Recuperado de: <http://www.agilemanifesto.org/>
- Beck, K., (2004). *Extreme programming explained: Embrace change* (2nd ed.). Boston, MA: Addison-Wesley Professional.
- David, J.-L., Gousset, M., & Gunvaldson, E. (2006). *Professional team foundation server*. Indianapolis, IN: Wiley Publishing
- Humphrey, W. S. (2005). *TSP: Leading a development team*. Upper Saddle River, NJ: Addison-Wesley Professional.
- Humphrey, W. S., Chick, T. A., Nichols, W., & Pomeroy-Huff, M. (2010). *Team software process (TSP) Body of knowledge (BOK)*. Pittsburgh, PA: Software Engineering Institute. <http://www.sei.cmu.edu/reports/10tr020.pdf>
- Mak, B. (2006, Abril). *Microsoft solution framework for agile software development* (slides). Presentación realizada en Microsoft solution framework for agile software development workshop. Recuperada de: <http://download.microsoft.com/download/8/6/3/86375d9e-1263-4ba0-b5f2-f696aceb94f3/20060418.ppt>
- Microsoft. (2010a). *MSF for agile software development v5.0*. Recuperado de:

- <http://msdn.microsoft.com/en-us/library/dd380647.aspx>
- Microsoft. (2010b). *Team foundation installation guide for visual studio 2010*. Recuperado de: <http://www.microsoft.com/download/en/details.aspx?id=24337>
- Microsoft. (2010c). *Visual studio team foundation administrator's guide*. Recuperado de: <http://www.microsoft.com/download/en/details.aspx?id=7330>
- Phillips, D. (2004). *The software project manager's handbook. Principles that work at work*. Los Alamitos, CA: IEEE Computer Society.
- Pino, F. J., García, F., Piattini, M., & Oktaba, H. (2006). Revisión sistemática de mejora de procesos de software en pequeñas y medianas empresas de software. *Revista española de innovación, calidad e ingeniería de software* (2)1. 6-23. <http://redalyc.uaemex.mx/pdf/922/92220103.pdf>
- Project Management Institute. (2008). *A guide to the project management body of knowledge (Pmbok Guide)* (4th ed.). Newtown Square, PA: Autor.
- Sawyer, S. (2004). Software development teams. *Communications of the ACM*, (47)12, 95-99. doi: 10.1145/1035134.1035140
- Schwaber, K. (2004). *Agile project management with Scrum*. Redmond, WA: Microsoft Press.
- Schwaber, K., & Beedle, M. (2001). *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.
- Sutherland, J. (2010). *Agile principles and values*. Recuperado de : <http://msdn.microsoft.com/en-us/library/dd997578.aspx>
- Traa, J. W. A. (2006). *Rational unified process vs. Microsoft solutions framework: A comparative study*. (Master of Informatics & Economics, Erasmus University Rotterdam, The Netherlands). (254014jt)



## **Currículum vitae**

### **Andrés Felipe Paz Loboguerrero**

Ingeniero de Sistemas y de la Universidad Icesi (Cali, Colombia) y estudiante de la Maestría en Gestión de Informática y Telecomunicaciones de la misma universidad. Se desempeña como catedrático en cursos de enseñanza de programación y como investigador del Grupo de investigación en informática y telecomunicaciones (i2t) de la Universidad Icesi. Entre sus áreas de interés están el desarrollo generativo de software, las arquitecturas de software y los procesos de software.

### **Lorena Castañeda Bueno**

Ingeniera de Sistemas e Ingeniera Telemática de la Universidad Icesi, estudiante de la Maestría en Gestión de Informática y Telecomunicaciones de la misma universidad. Se desempeña como profesora asistente y como investigadora del Grupo de investigación en informática y telecomunicaciones (i2t), de la misma universidad. Entre sus áreas de interés están la ingeniería de líneas de producto de software dirigidas por modelos y los sistemas de software auto-adaptativos y auto-gestionados.

### **Hugo Fernando Arboleda Jiménez, PhD.**

Ingeniero de Sistemas y Computación de la Universidad del Valle, Magíster en Ingeniería de Sistemas y Computación de la Universidad de Los Andes. Obtuvo el grado de Docteur en Informatique por la École des Mines de Nantes (Francia) en 2009 y el de Doctor en Ingeniería por la Universidad de Los Andes en 2010. Es profesor asociado en la Universidad Icesi y miembro del Grupo de Investigación en Informática y Telecomunicaciones (i2t), de la misma universidad. Entre sus áreas de interés están el desarrollo generativo de software, la ingeniería web dirigida por modelos, los procesos de software y las arquitecturas de software.