

# Metodología de Software MSF en pequeñas empresas

## MSF software methodology in small businesses

William Arévalo  
Magíster en Ingeniería  
Tecnológico de Antioquia, Institución Universitaria  
warevalo@tdea.edu.co

Andrés Atehortúa  
Ingeniero en Software  
Tecnológico de Antioquia, Institución Universitaria  
Aatehor2@tdea.edu.co

*Recibido: 15 de septiembre 2012  
Aprobado: 5 de noviembre 2012*

### Resumen

La Ingeniería de Software es una de las ramas de la tecnología, fundamentada en la prestación de servicios con altos estándares de calidad, para cumplir con los objetivos corporativos del interesado. Por eso se profundiza más en el ciclo de vida del software, esto le brinda al ingeniero de software una mayor capacidad de análisis de los problemas planteados, de acuerdo al contexto en el que se encuentra. En este artículo se presenta la metodología de desarrollo de software MSF, como una herramienta aplicada a pequeñas empresas, así como sus efectos dentro de la organización. Se demuestra el impacto de forma directa o indirecta, positiva o negativa en el mejoramiento continuo y la estandarización de procesos, de acuerdo con cada necesidad planteada por un usuario gerente o usuario final.

**Palabras clave:** Software, Metodología MSF, ciclo de vida

### Abstract

Software engineering is an area of technology, grounded on providing services with high quality standards, in order to fulfill a client's corporate goals. This is why, focus is made in software life cycle, allowing the software engineer to increase his/her ability to analyze problems according to his/her context. This paper presents the methodology of software development (MSF, Microsoft Solution Framework) as a tool applied to small businesses, along with its effects within an organization. Direct or indirect, and positive or negative impacts in ongoing process improvement and standardization —according to a given need from a manager user or end user— are shown.

**Keywords:** software, MSF methodology, life cycle.

## 1. Introducción

Según el Laboratorio Nacional de Calidad del Software del Instituto Nacional de Tecnologías de la Comunicación de España (2009), la metodología de software es un “conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo”. Lo anterior se complementa con la siguiente definición de la Real Academia Española (2012): “Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal”. Si se buscaran más definiciones seguramente se encontrarían miles que terminarían en lo mismo: conjunto de pasos integrados que permiten tener un resultado con altos niveles de calidad y posible satisfacción por parte del usuario final.

En relación con lo anterior también se puede percibir que existen varias metodologías, entre las cuales se encuentra la metodología Microsoft Solution Framework —MSF— (Microsoft, 2000), que permite tener control total sobre cada uno de los procesos del desarrollo de soluciones, y le brinda al ingeniero herramientas para realizar un mejor análisis del ciclo de vida de desarrollo que desea utilizar. Actualmente, y desde hace mucho tiempo, se utilizan ciclos de vida como: El modelo en cascada, el modelo en espiral, el modelo de prototipos (Pressman, 2005), entre otros. Normalmente se suele utilizar el modelo en espiral en muchas compañías, ya que permite la creación de soluciones de forma modular.

Por eso las necesidades se han incrementado a nivel de desarrollo, lo que ha dado paso al nacimiento de las metodologías de desarrollo, que a su vez utilizan procesos predefinidos enfocados a la generación de productos con excelente calidad. Con base en lo anterior surge la Metodología MSF, conjunto de pasos divididos en cinco etapas concebidas así: visión, planificación, desarrollo, estabilización y liberación.

Este artículo se desarrolla de la siguiente forma: en la sección 2 se muestra la metodología MSF de forma general, luego en la sección 3 se describe el conjunto de pasos que se deben tener en cuenta para que las pequeñas empresas implementen la metodología MSF, y por último en la sección 4 se presentan las conclusiones y el trabajo futuro.

## 2. Marco general del modelo MSF

Comprende un conjunto de modelos, conceptos y guías que contribuyen a alinear los objetivos de negocio y los tecnológicos, reducir los costos de la utilización de nuevas tecnologías y asegurar el éxito en la implantación de las tecnologías (Microsoft, 2000). Esta metodología comprende cinco etapas:

- \* Visión
- \* Planificación
- \* Desarrollo
- \* Estabilización
- \* Liberación

Por medio de estas etapas se puede concebir cualquier proyecto de software, adaptando el ciclo de vida que sea necesario para la solución del problema que se plantee (Microsoft, 2000).

El proceso de desarrollo de software, utilizando completamente esta metodología, puede generar entre 15 y 30 subprocesos, dependiendo de la complejidad de la solución y su impacto en el modelo de negocio de la compañía que lo emplee. Metodologías como RUP permiten tener soluciones robustas por medio de procesos complejos, los cuales están también dentro de un número alto de pasos (Microsoft, 2000).

Cuando se habla de un proceso de desarrollo basado de la metodología MSF, se especifica como un flujo del proceso la siguiente secuencia de pasos:

El usuario realiza una solicitud de una necesidad, la cual, en caso de ser aprobada por el comité primario, pasará a tener un acuerdo de entendimiento que posteriormente será entregado a la Project

management office —PMO—, u oficina de administración de proyectos. En este punto el proyecto se entrega a los gestores de planes quienes designarán un analista de soluciones para que le realice el debido control, la gestión y el seguimiento en las etapas de planeación, desarrollo, estabilización y liberación.

Entre los procesos que se utilizan en las empresas pequeñas se tiene que la PMO o las unidades gestoras de proyectos son áreas que generan sobrecostos por la cantidad de personal que se tiene que contratar para esto, de ahí que se utiliza la figura de analistas de software para que se encarguen de los proyectos encabezados por un *team leader* o líder de equipo.

### 3. Metodología MSF en pequeñas empresas

A simple vista con la metodología MSF, es imposible de utilizar cuando se desea realizar una solución de forma simple, sin que se afecte de forma abrupta el modelo de negocio de la compañía o el área de tecnología que la utiliza. Para ello se desarrollan las cinco etapas omitiendo algunos pasos, entre los cuales están:

- Entrega a la PMO
- Entrega a la unidad de planes



**Figura 1.** Fases de MSF *Microsoft Solution Framework*.

Y la reducción de espacios en los formatos, lo que permite que no se transgreda la metodología y se alcance a tener una solución con calidad y en poco tiempo. La Figura 1 presenta las fases de la metodología propuesta.

#### 3.1. Visión

En esta etapa el ingeniero de software debe realizar los siguientes procesos (ver la Figura 2), que permiten tener una idea clara del planteamiento del problema.

##### 3.1.1. Acuerdo de entendimiento

Documento donde se consigna: el área usuaria, el gerente usuario, el usuario responsable, el gerente de la solución (el ingeniero de software), el

impacto legal, la fecha límite de entrega, la fecha inicial, la fecha de actualización, la versión del documento, comentarios, la fecha de aprobación, la descripción de la situación actual (esta última debe de ser lo más detallada posible), la descripción de la necesidad y los resultados esperados (de forma detallada), la necesidad específica, los objetivos, la justificación, la información complementaria y las áreas organizacionales involucradas en la solución.

### 3.1.2. Levantamiento de requisitos

El levantamiento de requisitos debe de cumplir con los siguientes parámetros:

- Necesario
- No ambiguo
- Conciso
- Consistente
- Completo
- Alcanzable
- Verificable

Para el desarrollo de los requisitos se deben tener las siguientes variables como mínimo: tipo de requisito (funcional, no funcional), id, id necesidad, responsable de levantar el requisito, nombre del requisito, descripción y áreas impactadas.

### 3.1.3. Verificación de requisitos

Para la verificación de los requisitos se puede realizar un cuestionario, el cual se aplicará al usuario final que intervenga en el proyecto, de tal forma que permita ver el cumplimiento de los parámetros antes descritos.



Figura 2. Etapa Visión Metodología MSF

## 3.2. Planificación

Esta es una de las etapas con mayor impacto durante el proceso de desarrollo del software, ya que contiene gran parte del análisis del proyecto, se divide en:

### 3.2.1. Creación de casos de uso

El mínimo de parámetros requerido para la creación de un caso de uso en esta metodología que se utiliza es el siguiente: nombre del caso de uso, código del caso de uso, requisito al que se asocia, descripción detallada, diagrama, actores, descripción de las responsabilidades por cada actor que se haya nombrado, precondiciones, poscondiciones, flujo primario y flujos alternos.

### 3.2.2. Verificación de casos de uso

Para la verificación de los casos de uso se debe tener en cuenta que sean concisos, necesarios, consistentes, alcanzables, verificables y que no sean ambiguos. Todo esto acompañado del usuario gerente y el ingeniero de software, de tal forma que se puedan determinar posibles errores en la creación de los casos de uso.

### 3.2.3. Realizar diagrama de clases

Como en todo proyecto de software es importante tener una consistente definición del diagrama de

clases, ya que este le permitirá al equipo desarrollador de la solución tener una idea clara del problema y dará pautas exactas de lo que se debe hacer en relación con el código fuente.

### 3.2.4. Modelo entidad relación

Este es quizá uno de los pasos o procesos que más tiempo requiere, porque muchas veces los proyectos de software tienen que recibir cargas iniciales de otros aplicativos sin normalizar o con inconsistencias en los datos. Por ello el Modelo Entidad Relación debe tener una clara definición, pues esto permitirá que en el momento del desarrollo se evite duplicidad de la información, inconsistencias o fallos en la misma, y así brindar una mejor prestación del servicio que se desea dar.

### 3.2.5. Diagramas de actividades

En este proceso el ingeniero de software podrá dar mayores herramientas al equipo desarrollador, gracias a que el diagrama de actividades permite tener una visión clara de los flujos de los procesos que se van a sistematizar, en concordancia con los casos de uso, lo que lo convierte en un complemento de estos.

### 3.2.6. Diagramas de secuencia

Cuando el ingeniero en software realiza esta actividad, le dará al equipo desarrollador todas las herramientas necesarias para poder tener una so-

lución con un alto nivel de calidad y de detección de errores.

### 3.2.7. Diagrama de despliegue

Es importante saber que el ingeniero de software no solo se debe dedicar a realizar el proceso de análisis de las soluciones, sino que también es importante que tenga conocimientos de hardware y software para así realizar estudios que le permitan una mejor ejecución y cumplimiento de los objetivos propuestos dentro del acuerdo de entendimiento. Este diagrama es necesario al interior de las compañías, ya que le permitirá al equipo encargado de implementaciones tener control sobre los pasos y saber qué necesidades hay que satisfacer para que la liberación posterior de la solución sea un éxito.

### 3.2.8. Diagrama de procesos

Como es de conocimiento general de los ingenieros de software o analistas de software, toda solución que implique administración de la información deberá cambiar los procesos de una compañía, sin que estos afecten la operación de las empresas de gran forma. Sin embargo, es necesario realizar este proceso, ya que le permitirá a los ingenieros de procesos o a los encargados del área de calidad crear nuevas caracterizaciones de procesos o modificar las actuales.

El resumen de esta etapa se presenta en la Figura 3, donde se pueden visualizar de una forma más práctica los pasos.



Figura 3. Etapa Planificación, Metodología MSF

### 3.3. Desarrollo

Esta etapa de la metodología comprende la traducción de todo el análisis realizado en las etapas anteriores a código fuente, de tal forma que se pueda ver reflejado el primer resultado del software hacia el usuario final. El éxito de esta etapa depende de la rigurosidad y precisión con las que se realicen las etapas anteriores.

Para la selección de la herramienta de desarrollo el ingeniero debe analizar la herramienta que más se adecue a las necesidades de la empresa, teniendo en cuenta lenguajes de software libre que contribuyan a la disminución de costos, especialmente para pequeñas empresas que no cuentan con el capital suficiente para la adopción de herramientas bajo licencia de propietario.

### 3.4. Estabilización

La estabilización es el proceso en el cual se realizan todas las pruebas de caja blanca, caja negra, ruta crítica y saturación, para poder visualizar si

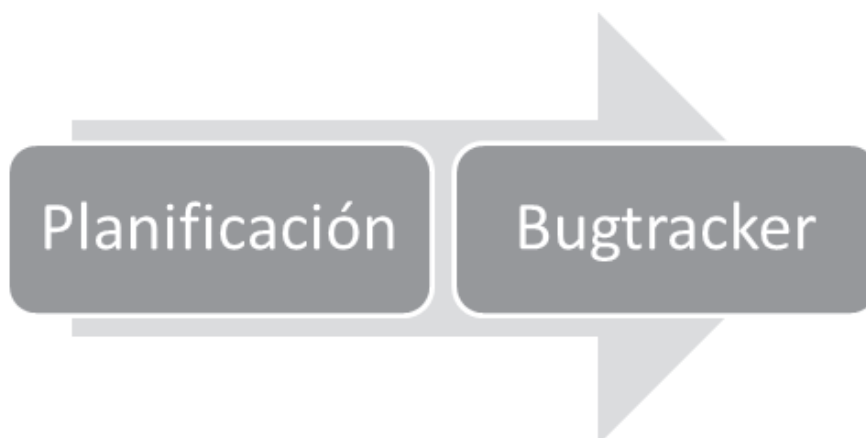
los resultados antes planteados por el equipo de desarrollo cumplen con la necesidad presentada. Es importante anotar que para una mayor y mejor estabilización, quien realice las pruebas de software no debe ser del equipo de desarrollo, pues esto hace que la solución pueda madurar de una mejor manera. De acuerdo con lo anterior se plantean los siguientes subprocesos para la estabilización de la solución (ver Figura 4).

#### 3.4.1. Planificación

Consiste en diseñar el documento con los casos de prueba que se ejecutarán.

#### 3.4.2. Bugtracker

Este documento es de gran importancia durante todo el proceso, ya que permite visualizar los posibles fallos que hacen que la operación de una compañía o que el desarrollo de un proceso se afecten negativamente. El Bugtracker permite tener el consolidado de errores que se encuentran en el sistema para retroalimentar al equipo de desarrollo y hacerles seguimiento a las correcciones.



**Figura 4.** Etapa Estabilización, Metodología MSF

### 3.5. Liberación

La Liberación incluye procesos en los que se debe tener en cuenta el personal al que se le entregará el software y el personal que se encargará de las actualizaciones en el área de calidad de la compañía (ver

la Figura 5). Asimismo, el personal de infraestructura y seguridad, para el desarrollo de esta etapa, debe tener claridad en los siguientes pasos:

### 3.5.1. Analizar nuevos procesos

Si la compañía es pequeña pero cuenta con un área de calidad, es necesario realizar este paso, donde se encontrarán el ingeniero de software con el ingeniero de procesos o analista de calidad de la empresa y el usuario final. Por medio de una reunión, se diligencia un documento que debe tener como requisitos mínimos los siguientes parámetros: verificación de nuevos procesos o el rediseño del mismo; verificación de impacto en actividades dentro de los procesos; verificación de la creación y eliminación de recursos humanos; efectos contables; efectos ante entes de control, y verificación de realización de capacitación.

### 3.5.2. Análisis de personal extra

En este proceso el personal descrito en el numeral anterior deberá asumir y argumentar de forma escrita y detallada por qué es necesario recurso humano adicional en la organización, o cómo se puede acomodar el proceso nuevo para que su ejecución se lleve a cabo con los recursos existentes en la compañía.

### 3.5.3. Matriz de impacto

La elaboración de esta matriz permite verificar hasta qué punto la liberación de la solución puede perjudicar la operación de toda la compañía. Para su realización se deben tener en cuenta los siguientes parámetros: Interrupción total o parcial del servicio prestado al cliente por parte de la compañía;

nivel en el que se degrada alguno de los servicios prestados y los efectos en la operación de esta degradación; tipo de liberación (nueva solución o funcionalidad extra); forma en la que se afectan los componentes de hardware de otros servicios, y qué componentes nuevos ingresan a la infraestructura de la compañía.

### 3.5.4. Lista de chequeo de contingencias

Dentro de todo proceso de software, cuando se va a realizar la liberación de un producto, se debe analizar qué contingencias se tendrán en cuenta para poder llegar a una finalización exitosa. Para esta metodología los parámetros requeridos son: plataforma de contingencia local; plataforma de contingencia remota; procedimiento de activación y retorno de contingencia; esquema de replicación de la base de datos; esquema de replicación de objetos, y homologación de ambientes de producción con los de contingencia.

### 3.5.5. Capacitar al usuario final

Por medio de este proceso se finaliza todo el ciclo de la metodología, lo que permite al usuario final tener acceso a la solución de forma productiva. Siempre que se realice una capacitación al usuario final, se debe llevar registro escrito del evento, de este modo se puede asegurar una apropiada divulgación del conocimiento para una utilización óptima de la solución desarrollada.



Figura 5. Etapa Liberación, Metodología MSF



## 4. Conclusiones y trabajos futuros

Las metodologías de desarrollo de software permiten crear soluciones con altos estándares de calidad que generan grandes oportunidades de valor agregado en las compañías, pues sirven para mejorar tiempos de respuesta, costos y oportunidades de capacitación al recurso humano.

La aparición de la metodología MSF genera la necesidad de verificar la documentación de la misma. Teniendo en cuenta que cuando se realizan implantaciones basadas en esta metodología de forma completa, se puede llegar a un ciclo demasiado grande de procesos que pueden llevar a que una empresa pequeña sobredimensione las funciones de cada empleado. Esto ocasiona un problema ambiental dentro de la compañía, ya que transgrede el concepto de valor agregado.

Al analizar los procesos de una pequeña empresa, se encontró que se puede disminuir esta metodología de 35 pasos a 18. Con esto no se afirma que estas fases no aporten al proceso, solo que lo hacen más robusto para una pequeña empresa. Esto garantizó que se cumplieran unos requisitos mínimos que aportaran calidad al producto, valor agregado y bajos costos, y se fortaleciera el sistema de gestión de calidad de cada cliente o empresa.

Pese a estas conclusiones, aún se debe ahondar en el estudio para la caracterización del proceso de

desarrollo de softwares que disminuyan impactos negativos en el uso de las TIC en pequeñas empresas, donde se tengan en cuenta las características de este tipo de empresas.

## Agradecimientos

Este artículo se realizó con el apoyo de la Federación Nacional de Cafeteros de Colombia, el Comité Departamental de Cafeteros de Antioquia y Pelope Consultoría.

## Referencias

Laboratorio Nacional de Calidad del Software del Instituto Nacional de Tecnologías de la Comunicación de España (2009). *Ingeniería de Software: Metodologías y ciclos de vida*. Madrid, España.

Microsoft (2000). *Microsoft Solutions Framework (MSF): Disciplinas y buenas prácticas para el desarrollo e implantación de proyectos*. Microsoft Colombia. Disponible en: <http://www.microsoft.com/colombia/portafolio/msf.htm> [Consultado, noviembre de 2012].

Pressman, R.S. (2002). *Ingeniería de Software un enfoque práctico*. (5.ª ed.). Adaptado por Darrel Ince. Madrid, España.