

Metodología de Desarrollo para la Mejora de Procesos de Software a Través del Desarrollo Dirigido por Pruebas

¹Centeno-Téllez A. ²Gómez-Gil P.

Resumen

El presente trabajo describe una nueva metodología Ágil para la mejora de procesos de software llamada DEPYM. Ésta metodología de desarrollo fue creada para ser usada en el proceso de Desarrollo y Mantenimiento de software de Moprosoft y está basada en el desarrollo dirigido por pruebas. La metodología serviría para que los desarrolladores de software de pequeñas empresas mexicanas aprovechen la agilidad del desarrollo dirigido por pruebas sin sacrificar la calidad solicitada por las normas mexicanas. El artículo incluye las bases teóricas sobre las que se basa el proyecto así como los resultados obtenidos a la fecha.

1. Introducción

Los problemas presentes en las empresas de la construcción de software mexicanas están íntimamente relacionados con la falta de procesos maduros y repetibles usados en la elaboración de sus productos. Esta falta de procesos institucionalizados se refleja en la mala calidad de los productos, entregas fuera de calendario y presupuestos rebasados.

Las Metodologías Ágiles son un ejemplo de prácticas específicas para desarrollar software. La programación extrema (*Extreme Programming*, XP) es una disciplina de desarrollo de software basada en la simplicidad, comunicación, retroalimentación y entregas frecuentes. XP utiliza prácticas sencillas a ser utilizadas por el equipo de desarrollo y con suficiente retroalimentación para motivar al equipo a conocer el estado real del proyecto [1]. El desarrollo dirigido por pruebas (*Test Driven*

Development, TDD) es una de las 12 prácticas claves de la Programación Extrema. En TDD los desarrolladores de software "prueban primero, después codifican," enfocando-se inicialmente en la verificación y validación de los requerimientos de software, mediante la construcción de pruebas unitarias automatizadas. En este paradigma, el diseño evoluciona como un nuevo código que es escrito para satisfacer la pruebas que fallaron [2]. La figura 1 muestra el modelo de procesos del desarrollo dirigido por pruebas.

Moprosoft es el modelo de procesos para la industria de software mexicana que fue creado para apoyar en la solución de la problemática de la crisis de software [4]. El objetivo de Moprosoft es facilitar a las organizaciones dedicadas al desarrollo y mantenimiento de software la adopción de las mejores prácticas reconocidas internacionalmente a través de modelos como: SW-CMM, CMMi,

¹Universidad Veacruzana, USBI Campus Ixtac, Ixtaczoquiltán Veracruz México, acenteno@uv.mx

²Coordinación de Computación, Instituto Nacional de Astrofísica, Óptica y Electrónica. Tonantzintla, Puebla. pgomez@inaoep.mx, pgomez@acm

PSP, TSP, ISO/IEC 15504, PMBOK y SWEBOK.

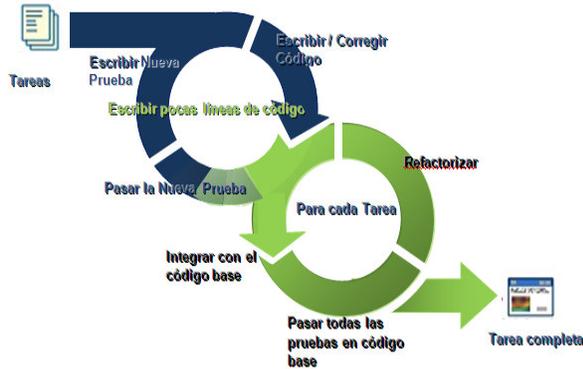


Figura 1.- Modelo de procesos de TDD [3].

De acuerdo a la norma Mexicana Moprosoft los procesos para la creación de software se agrupan en 3 categorías: Alta Dirección, Gestión de Procesos y Nivel Operativo [4]. Los procesos que define Moprosoft para cada categoría son:

Alta Dirección: Gestión de Negocio

Gestión: Gestión de Procesos, Gestión de Proyectos, Gestión de Recursos (Recursos Humanos y Ambiente de Trabajo, Bienes Servicios e Infraestructura, Conocimiento de la organización)

Operación: Administración de Proyectos Específicos, Desarrollo y Mantenimiento de Software.

Hasta Junio del 2009 se habían registrado 138 empresas evaluadas en algún proceso de calidad en 20 estados de la República Mexicana tales como: Aguascalientes, Baja California, Chihuahua, Coahuila, Colima, DF, Hidalgo, Jalisco, México, Michoacán, Nuevo León, Oaxaca, Puebla, Querétaro, Sinaloa, Sonora, Tlaxcala, Veracruz, Yucatán y Zacatecas [5]. La tabla 1 muestra la distribución de las empresas mexicanas evaluadas en 3 normas de calidad, donde puede notarse

que existen 90 empresas de las 138 evaluadas que han sido aprobadas en la norma mexicana NMX-I-059/02-NYCE-2005, lo que constituye el 65.22% del total nacional evaluado. De las 90 empresas Moprosoft, 86 se encuentran solamente en el nivel 1, lo que representa el 63.32% de las empresas evaluadas. Entonces el 62.32% de empresas mexicanas cuentan con procesos mínimos requeridos para terminar un proyecto de software, y no consideran aún aspectos más formales de calidad, costos y tiempos de entrega.

Tabla 1. Empresas Certificadas en Modelos de Calidad a Junio 2009 [5].

Modelo/Nivel	1	2	3	4	5	Total
CMM	0	2	6	2	4	14
CMMi	0	14	13	2	7	36
Moprosoft	86	3	1	0	0	90

2. Hipótesis y objetivos

Nuestra hipótesis de investigación es:

“Es posible mejorar la calidad de los productos de software de las empresas mexicanas que cuenten con el nivel 1 de la verificación Moprosoft usando un modelo de desarrollo dirigido por pruebas en el proceso de desarrollo y mantenimiento de software”.

El objetivo principal de este trabajo es crear un modelo de desarrollo de software, llamado DEPYM (Metodología de Desarrollo Dirigido por Pruebas y Mejora de procesos para PyMEs), basado en la técnica de desarrollo dirigido por pruebas que permita a empresas mexicanas con Moprosoft Nivel 1 mejorar su desempeño.

Los objetivos específicos del trabajo son los siguientes:

1. Determinar las fortalezas y debilidades de las empresas con nivel 1 Moprosoft,

con respecto a su metodología de desarrollo de software,

2. Crear el modelo DEPYM y definir su estrategia de implementación, a fin de que potencialice las fortalezas de las empresas.

3. Metodología de Investigación

Para alcanzar los objetivos definidos, se están realizando los pasos descritos en la figura 2. A la fecha se está realizando el tercer paso (diseño del método de implementación del modelo).

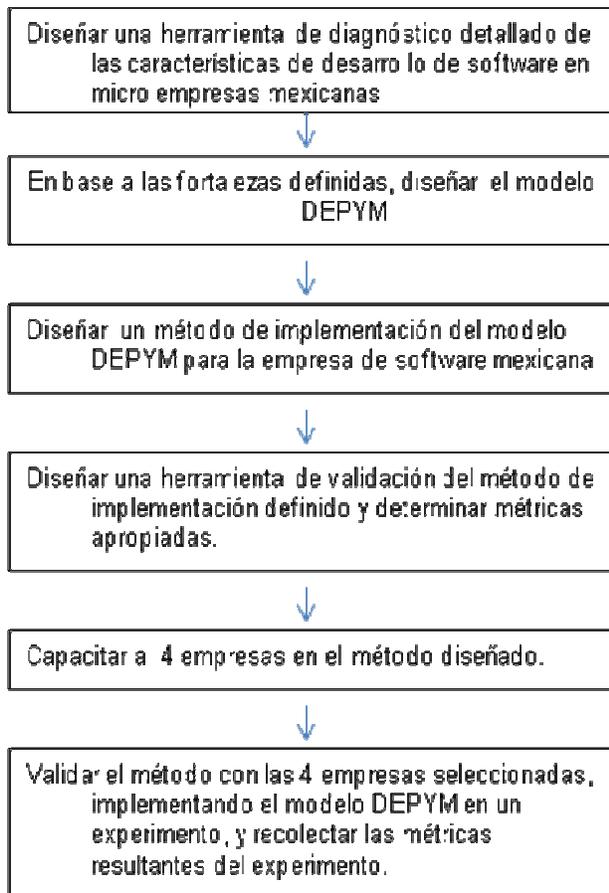


Figura 2.- Metodología de investigación [8].

4. La Metodología DEPYM

La Metodología de Desarrollo Dirigido por Pruebas y Mejora de procesos para PyMEs toma las ventajas del framework TDD definido por Kent Beck [6] y se

ajusta a las especificaciones del modelo de procesos Moprosoft en su proceso de Desarrollo y mantenimiento de software (DMS).

La figura 3 muestra el modelo de procesos de DEPYM. La entrada a los procesos de DEPYM es el “Plan de Desarrollo,” que es provisto por el proceso Administración de Proyectos Específicos de Moprosoft; los artefactos de salida son los mismos que para el modelo tradicional Moprosoft. Cabe mencionar que los productos de salida marcados con “*”, tales como “Plan de Pruebas de Sistema”, “Reporte de Pruebas del Sistema”, “Plan de Pruebas de Integración” y “Reporte de Pruebas de integración”, entre otros, pertenecen al nivel 2 de Moprosoft. Por tanto nuestro modelo DEPYM con su enfoque dirigido por pruebas apoyará de forma sustancial a las empresas a alcanzar fácilmente el nivel 2 en el proceso DMS, lo cual es altamente deseable para empezar a preparar a las empresas para una competencia internacional. La Tabla 2 muestra los productos de trabajo resultante del modelo DEPYM.

Tabla 2. Productos de trabajo resultantes de la metodología DEPYM [8]

1. Configuración de Software
 - a. Especificación de requerimientos
 - b. Documento de análisis y diseño
 - c. Registro de Rastreo *
 - d. Plan de Pruebas de Sistema *
 - e. Reporte de Pruebas de Sistema *
 - f. Plan de Pruebas de integración *
 - g. Reporte de Pruebas de integración *
 - h. Manual del usuario
 - i. Manual de operación
 - j. Manual de mantenimiento *
2. Reporte de actividades *
3. Lecciones aprendidas *
4. Reporte de mediciones y sugerencia de mejoras *

DEPYM contiene los siguientes procesos:

1. Procesos la obtención del Project Charter a partir del “Plan de Desarrollo.” En este documento se define la descripción del proyecto, los desarrolladores involucrados, las entregas (*releases*) y las historias de usuario.
2. Procesos para realizar la definición de las historias de usuario, las cuales representan requerimientos funcionales o necesidades específicas del cliente. Asimismo, se incluye una definición de prioridades, asignación de responsables, cálculo del valor ganado y estimación de tiempos.
3. Procesos para detallar las historias de usuario con sus respectivas tareas, prioridades, estimación de tiempo en horas y valor planeado para cada tarea.
4. Procesos para la generación de las pruebas unitarias. En este paso se utiliza el patrón *Arranging-Act-Assert* [7]
5. Procesos para la generación de código a partir de las pruebas estableciendo las mejores prácticas del enfoque TDD.
6. Procesos para las pruebas de verificación funcional y de integración.

Como se mencionó anteriormente, el *Project charter* es el documento de visión y alcance que permite definir información general sobre el proyecto tal como el nombre, descripción, estimación de fechas de arranque y terminación, *stakeholders*, *releases* y las historias de usuario en las que se dividirá el proyecto. Además este documento

permite definir cuál es el estándar de codificación que se usará para construir los programas. De igual forma aquí se define cual será el documento de configuración de software que ayudará al nombrado de carpetas, archivos, formatos de fecha y horas, entre otros. La figura 4 muestra un ejemplo de la sección de historias de usuario del modelo DEPYM para un proyecto de denominado “Cálculo de Nomina.”

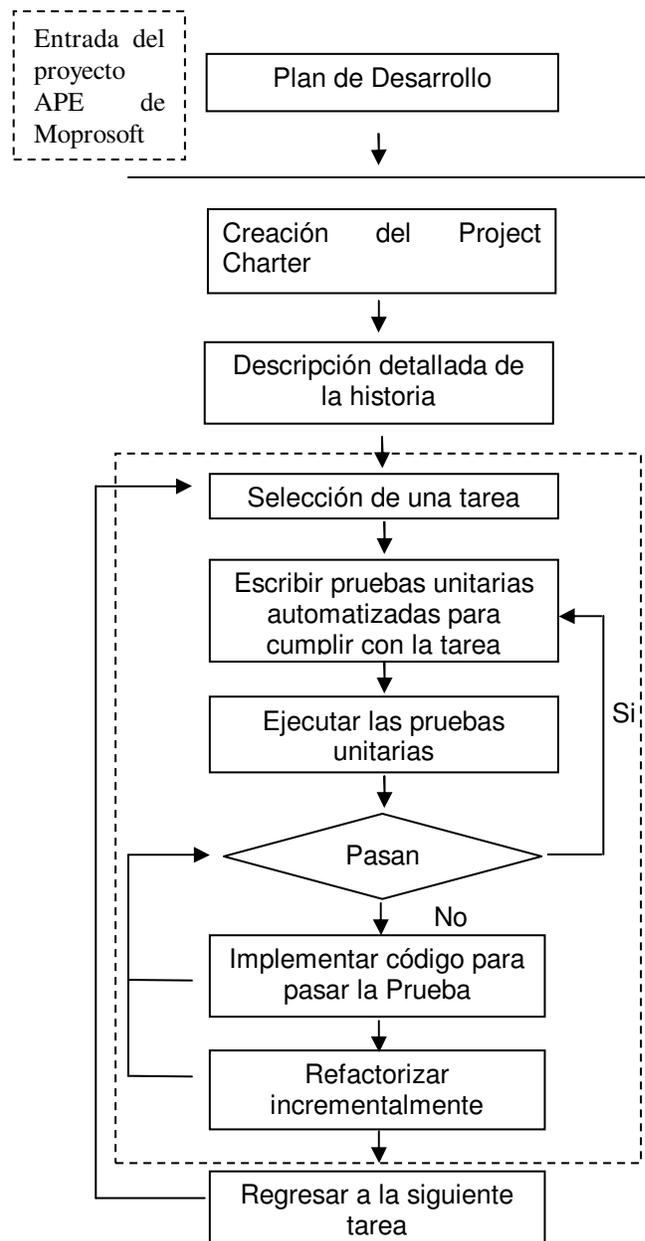


Figura 3.- Modelo de procesos de la metodología DEPYM para una tarea correspondiente a una historia de usuario del proyecto.

El *Project charter* permite determinar el estado actual del proyecto a través del valor ganado obtenido por cada una de las historias de usuario. Las historias de usuario permiten definir de forma modular cada uno de los requerimientos funcionales del proyecto. Las historias determinan a detalle cada una de las tareas necesarias para llevar a cabo una historia, permitiendo además requerimientos específicos e información adicional sobre los procesos. La figura 5 muestra el formato para definir una historia de usuario en el modelo DEPYM.

ID	User Stories	Prioridad	Release	Resp.	Tiempo	Val Plan	Val Gan.
login	Login	3	3	ACT	1	10	5
user	ABC usuarios	3	3	ACT	2	10	0
mast	Master Page	1	1	RRA	2	10	0
imp	Importar datos de Excel	2	2	RRA	1	20	0
rep	Generar el reporte de nomina	1	1	IAP	6	40	35
capt	Captura manual de movimientos	2	2	IAP	3	10	0
	Total Valor ganado del Proyecto	40					
	Porcentaje de avance del proyecto	40%					

Figura 4.- Sección de historias de usuario de un Project Charter.

El modelo DEPYM está totalmente orientado al desarrollo dirigido por pruebas, por tanto para construir el código de una historia de usuario, se definen sus tareas y de cada tarea se definen de forma detallada cada una de las pruebas unitarias que se deberán cumplir para terminar esa tarea. Para la definición de las pruebas de usuario se usa el patrón *Arranging-Act-Assert*. La figura 6 muestra el formato para la definición de las pruebas unitarias de una tarea seleccionada.

Tasks						
ID	Task	Prioridad	Fecha creación	Tiempo	Val plan	Val.gan
Interf	Construir la interfaz para mandar a procesar al reporte	3	13-02-10	4	20	0
Repeco	Construir el reporte concentrado del 2% sobre nomina	2	9-02-2010	8	40	0
Repede	Construir el reporte detallado del 2% sobre nomina	1	11-02-2010	8	40	0

Figura 5.- Tareas de una historia de usuario seleccionada del Project charter de la figura 4.

5. Contribuciones

Las contribuciones de la investigación son las siguientes:

- Creación una metodología Ágil de desarrollo de software adaptable las empresas mexicanas y basado en las mejores prácticas de los métodos Ágiles
- Procesos y formas adaptables a empresas pymes y a Moprosoft
- Definición de procesos y plantillas para la definición de la carta del proyecto
- Procesos y definición de la forma de historia de usuario
- Procesos y definición de la forma de la tarea usando el patrón *Arranging-Act-Assert*.

- Primera versión de la metodología DEPYM.
- Caracterización de las empresas PYMES de desarrollo de software.
- Determinación del dominio de aplicación de la metodología.

Actualmente se está refinando el modelo. Está pendiente el diseño de la herramienta de validación del modelo y los experimentos utilizando dicho modelo.

Nombre Prueba Unitaria: testImpresionReporte	Descripción: impresión del reporte detallado vía web
<pre>[TestMethod] void testImpresionReporte() { Arranging ReporteDetallado rptDetallado= new ReporteDetallado(); Act [Aquí van las operaciones] rptDetallado.procesarReporte(100); Assert: [Aquí va el assert que va a validar la prueba] Assert.IsNotNull (rptDetallado) }</pre>	

Figura 6.- Ejemplo de una prueba de una tarea seleccionada.

6. Resultados actuales

Al momento de la publicación de este artículo, se tienen los siguientes avances en la investigación:

7. Referencias

- [1] Ruvalcaba Mara, Software Guru Edición Enero-Febrero 2005, México Pág. 4.
- [2] Jones Christopher G. "Test-driven development goes to school." CCSC: Rocky Mountain Conference. Oct. 2004
- [3] Nichiappan Nagappan et al "Realizing quality improvement through test driven development: results and experiences of four industrial teams." Empirical Software Engineering, Vol. 13, pp. 289-302.
- [4] Gómez-Gil Pilar, "MOPROSOFT: Un Camino Hacia el Éxito Mundial en el Desarrollo del Software Mexicano." Memorias de la Semana de Informática 2007. Instituto Tecnológico de Puebla, Oct. 2007.
- [5] Sistema Nacional de Indicadores de la Industria de TI (SNIITI), Empresas Certificadas en Modelos de Calidad, fecha de consulta 16.04.09, disponible en: <http://www.edigital.economia.gob.mx/info/agent.aspx?docid=14>
- [6] Beck Kent, "Test Driven Development by example." Addison Wesley, Pág. 6-10

[7] [Conery, 2009] Conery, Guthrie, Haak, Hanselman. Professional ASP NET MVC 1.0, 2009.

[8] Centeno, Adolfo. Propuesta de Proyecto Doctoral en Ingeniería de Software. Centro interdisciplinario de Posgrado. Universidad Popular Autónoma del Estado de Puebla, México. 2009