

Evaluación de calidad en el desarrollo de software dirigido por modelos

Quality evaluation in software development model driven by models

Viviana Esterkin¹ Claudia Pons^{1,2*}

Recibido 21 de septiembre de 2015, aceptado 8 de septiembre de 2016

Received: September 21, 2015 Accepted: September 8, 2016

RESUMEN

El Desarrollo de Software Dirigido por Modelos (MDD) es una disciplina que está generando muchas expectativas como alternativa a los métodos convencionales de producción de software. Dado que MDD es un paradigma emergente, aún no se han establecido estándares para medir la calidad de sus aplicaciones. Este trabajo ofrece un aporte en este sentido, realizando un análisis de las buenas prácticas MDD en relación con el nivel de madurez 2 del CMMI-DEV 1.3. Para cada práctica específica en cada Área de Proceso del Nivel 2 del CMMI-DEV 1.3, las mejores prácticas MDD fueron analizadas para determinar si brindan soporte a cada práctica específica. Posteriormente, se procedió a validar los resultados obtenidos consultando a profesionales de ingeniería de software especialistas en el tema. Para cada área de proceso, el grado de soporte brindado por MDD para cada práctica específica fue calculado. Finalmente se elaboraron propuestas que permitirían incrementar el soporte MDD, con vistas a lograr que una organización que lo utilice esté en condiciones de certificar CMMI-DEV 1.3 Nivel 2.

Palabras clave: Modelo de madurez, CMMI-DEV 1.3, desarrollo de software dirigido por modelos (DSDM), evaluación de calidad.

ABSTRACT

Software Development Process by Model-Driven (MDD) is a discipline that is generating a lot of expectations as an alternative to conventional methods of software production. Given that MDD is an emerging paradigm, standards for measuring the quality of its applications have not been established yet. This paper provides a contribution in this regard, analyzing MDD good practices in relation to CMMI-DEV 1.3 Level 2. For each specific practice in each CMMI level 2 Process Area, MDD best practices were assessed to determine whether they support or not each CMMI Level 2 specific practice. Software engineering professionals were consulted to evaluate the results. For each Process Area the percentage of specific practices supported by MDD was calculated. The assessment resulted in the elaboration of proposals to enhance MDD support in order to certify CMMI-DEV 1.3 Level 2.

Keywords: Capability Maturity Model, CMMI-DEV 1.3, Model Driven Software Development (MDD), Quality evaluation.

¹ Facultad de Tecnología Informática. Universidad Abierta Interamericana. Ciudad de Buenos Aires, Argentina.
E-mail: Viviana.esterkin@uai.edu.ar

² Comisión de Investigaciones Científicas CIC. Ciudad de Buenos Aires, Argentina. E-mail: Claudia.pons@uai.edu.ar

* Autor de correspondencia

INTRODUCCIÓN

El Desarrollo de Software Dirigido por Modelos (MDD, por su acepción en inglés “Model-Driven Software Development”) [1-2] es una disciplina que está generando muchas expectativas como alternativa sobresaliente a los métodos convencionales de producción de software. MDD plantea una nueva forma de entender el desarrollo y mantenimiento de sistemas de software con el uso de modelos como principales artefactos del proceso de desarrollo. En MDD, los modelos son utilizados para dirigir las tareas de comprensión, diseño, construcción, pruebas, despliegue, operación, gestión, mantenimiento y modificación de los sistemas. Una gran cantidad de trabajos teóricos y prácticos acompañan a este movimiento. Existen también herramientas que lo hacen ya realidad a nivel comercial, con numerosos ejemplos de casos exitosos de introducción del MDD en diferentes organizaciones, como puede verse en las recopilaciones de experiencias realizadas por D. Di Ruscio, R. F. Paige y A. Pierantonio [3] y por el Object Management Group, OMG [4].

Por otra parte, las certificaciones de calidad permiten testificar la eficacia y eficiencia de los procesos. La mayoría de las certificaciones tienen un impacto en el interior de las organizaciones, ya que, por un lado, obligan a pensar en las mejores formas de alcanzar objetivos antes de certificar y, por otro, a actuar de manera previsible luego, basando las decisiones en información cierta. Las certificaciones de calidad agregan valor al producto, previsibilidad al trabajo, y confianza a los clientes, lo que conlleva a aumentar la competitividad de la empresa. Se percibe como valioso el que la empresa se preocupe por dar a conocer su calidad y abra las puertas a mostrar cómo trabaja ante organismos de certificación externa e inclusive internacional.

Dado que MDD es un paradigma emergente, aun no se han establecido estándares para medir la calidad de sus aplicaciones. Este trabajo ofrece un aporte en este sentido, realizando un análisis de las buenas prácticas de MDD en relación con el estándar de facto para evaluación de calidad en procesos de desarrollo de sistemas de software.

Modelos de Madurez de Capacidades

El modelo de madurez de capacidades denominado CMMI (por las siglas en inglés para Capability

Maturity Model Integration) [5] es la integración de un grupo de modelos para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas. Provee una guía para aplicar una colección de buenas prácticas a estos procesos. Este modelo es administrado por el Instituto de Ingeniería de Software (SEI, por sus siglas en inglés para “Software Engineering Institute”) de la Universidad Carnegie Mellon; es considerado el estándar de facto para evaluación de calidad en procesos de desarrollo de sistemas de software. El CMMI tiene dos representaciones, por etapas y continuo, que permiten a la organización perseguir diferentes objetivos de mejora. La presentación y organización de la información es diferente para cada una, sin embargo el contenido es el mismo. En este estudio, se utilizará CMMI para desarrollo versión 1.3 (CMMI-DEV) [5].

El CMMI tiene cinco niveles de madurez, que indican cada uno, el nivel de madurez al que ha llegado una organización en el desarrollo de los procesos de software. Los niveles de madurez, se utilizan para describir un camino de evolución, recomendado a una organización que desea mejorar los procesos que utiliza para desarrollar productos o servicios. Los niveles de madurez caracterizan la mejora de una organización relativa a un conjunto de áreas de proceso. Un área de proceso es un conjunto de prácticas relacionadas de dicha área que, cuando se implementan colectivamente, satisfacen un conjunto de objetivos que se consideran importantes para lograr su mejora. Los cinco niveles de madurez son:

- **Inicial.** Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El resultado de los proyectos es impredecible.
- **Gerenciado.** En este nivel las organizaciones disponen de prácticas institucionalizadas de gestión de proyectos, existen métricas básicas y un razonable seguimiento de la calidad.
- **Definido.** Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos.

- **Gestionado.** Las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos.
- **Optimizado.** La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Desarrollo de software Dirigido por Modelos

El desarrollo dirigido por modelos es un enfoque para el desarrollo de software en el que los artefactos primarios de software son los modelos, a partir de los que se genera el código y otros artefactos [1-2]. MDD propone resolver los problemas actuales de desarrollo de software, utilizando un marco de trabajo que asegura portabilidad, interoperabilidad, independencia de plataforma y productividad [1]. Además, la arquitectura dirigida por modelos (MDA, por sus siglas en inglés para “Model-Driven Architecture”) [6] es un estándar propuesto y patrocinado por el Object Management Group (OMG) [7], concebido para dar soporte al desarrollo dirigido por modelos. MDA es una arquitectura que proporciona un conjunto de guías para estructurar especificaciones expresadas como modelos. Usando la metodología MDD/MDA, la funcionalidad del sistema será definida en primer lugar como un modelo independiente de la plataforma (Platform-Independent Model o PIM) por medio de un lenguaje específico para el dominio del que se trate. El modelo PIM puede traducirse entonces a uno o más modelos específicos de la plataforma (Platform-Specific models o PSMs) para la implementación correspondiente. La traducción desde el PIM hacia los PSMs se realiza normalmente utilizando herramientas automatizadas de transformación de modelos.

El MDD tiene un efecto profundo sobre los procesos con los que las aplicaciones de software son construidas. Las organizaciones y los proyectos frecuentemente dependen de expertos clave, quienes toman las decisiones respecto al sistema. MDD permite capturar su experiencia en los modelos y en las transformaciones, permitiendo de esta forma que otros miembros del equipo puedan aprovecharla sin requerir su presencia. Además, este conocimiento se mantiene aun cuando los expertos se alejen de la organización. Por otra parte, el costo de desarrollo y de testing se reduce significativamente al automatizar

gran parte del trabajo de generación del código y otros artefactos a partir de los modelos. mediante la automatización MDD favorece la generación consistente de los artefactos reduciéndose la presencia de errores.

Para lograr los beneficios de MDD el proceso de desarrollo de software debe adaptarse. El administrador de un proyecto MDD en realidad debe controlar un proyecto dentro de otro proyecto. El proyecto interno consiste en fabricar herramientas MDD para que sean usadas por el equipo que trabaja en el desarrollo del proyecto externo. El hecho de tener dos proyectos acoplados obliga a organizar y planificar más cuidadosamente, especialmente al comienzo del proyecto. A las dificultades usuales asociadas con cualquier proyecto de desarrollo, ahora debe sumarse un conjunto adicional de dependencias internas. Las herramientas MDD requeridas deben ser identificadas, desarrolladas e instaladas antes de que los desarrolladores las necesiten. Esto implica que los flujos de las tareas de ambos proyectos están interrelacionados.

Evaluando calidad en procesos MDD

Este trabajo tiene por objetivo identificar el soporte que MDD brinda al nivel de madurez 2 del CMMI-DEV. Para lograr dicho objetivo se analizan las siete Áreas de Proceso de Nivel 2, luego cada área se describe en términos de prácticas específicas, que al implementarse conducen a satisfacer sus objetivos. Las siete áreas son las siguientes:

- Gestión de la Configuración (en inglés, Configuration Management, CM),
- Gestión de los Acuerdos con Proveedores (en inglés, Supplier Agreement Management, SAM),
- Gestión de los Requerimientos (en inglés, Requirements Management, REQM),
- Aseguramiento de la Calidad del Proceso y del Producto (en inglés, Process and Product Quality Assurance, PPQA),
- Medición y Análisis (en inglés, Measurement and Analysis, MA),
- Monitoreo y Control del Proyecto (en inglés, Project Monitoring and Control, PMC),
- Planeamiento del Proyecto (en inglés, Project Planning, PP).

Este artículo está organizado de la siguiente forma: en primer lugar, se describe la selección de las

buenas prácticas de MDD propuestas en la literatura. Luego, en base a dicha selección, se analiza si MDD brinda soporte a las prácticas específicas definidas por CMMI-DEV 1.3 Nivel 2. Se analiza cada una de las Áreas del Nivel 2, evaluando el soporte que MDD brinda para cada una de sus prácticas específicas. Posteriormente, se procede a validar los resultados obtenidos consultando a profesionales de ingeniería de software especialistas en el tema. Finalmente se concluye cuáles son las Áreas de Proceso soportadas por MDD y cuál es el grado de soporte para cada una de ellas. La evaluación finaliza con la elaboración de propuestas que permitirían incrementar el soporte MDD, con vistas a lograr que una organización que lo utilice esté en condiciones de certificar CMMI-DEV 1.3 Nivel 2.

SELECCIÓN DE BUENAS PRÁCTICAS EN MDD

Para seleccionar las “buenas prácticas” del MDD se han tenido en cuenta los puntos de vista de autores especialistas en el tema, presentados en el artículo escrito por P. Swithinbank, M. Chessell, T. Gardner, C. Griffin, J. Man, H. Wylie y L. Yusuf [8], también en el artículo de C. Pons, R. Giardini y G. Pérez [9] y en la publicación de E. Ríos, T. Bozheva, A. Bediaga, N. Guilloureau, A. Rensink y J. Warmer [10]. Cada práctica seleccionada es identificada con la sigla BP (Buena Práctica) y un número (comenzando por el número 1). A continuación se enumeran las prácticas seleccionadas, agrupadas por su procedencia.

Las siguientes prácticas fueron extraídas del artículo de P. Swithinbank, M. Chessell, T. Gardner, C. Griffin, J. Man, H. Wylie y L. Yusuf [8]:

- **BP1. Identificar patrones comunes y estándares.** El arquitecto de soluciones identifica los patrones que se repiten en la aplicación de negocios. Estos patrones aparecen muchas veces debido al uso consistente de un estilo de arquitectura o debido a los requerimientos de las plataformas de ejecución.
- **BP2. Identificar activos MDD reusables.** En esta tarea el arquitecto de soluciones compara los patrones comunes que se identificaron en la tarea BP1, con los activos MDD existentes, haciendo los ajustes necesarios a su arquitectura, para explotar lo que ya está disponible. Los activos pueden provenir de proyectos MDD previos o bien de elementos estándar.
- **BP3. Definir el modelo de diseño.** El arquitecto de soluciones elige el tipo de modelo UML apropiado para los desarrolladores de la aplicación, para ser usado cuando se definan los detalles específicos de la componente que están construyendo. También crea una lista inicial de estereotipos para el perfil del UML.
- **BP4. Identificar el modelo independiente de la plataforma.** Este modelo puede ser realizado por el arquitecto de soluciones o bien por un desarrollador experimentado que comprenda los entornos de ejecución.
- **BP5. Producir los artefactos de muestra para los escenarios clave.** Un programador de aplicaciones codifica en forma manual los artefactos que van a actuar como planos detallados para plantillas y transformaciones.
- **BP6. Definir la cadena de herramientas MDD.** La tarea identifica las herramientas MDD necesarias para el desarrollo del proyecto. Una vez completada esta tarea, es posible crear un plan detallado del esfuerzo requerido para construir la cadena de herramientas MDD.
- **BP7. Validar la cadena de herramientas.** Esta tarea es realizada por el arquitecto de soluciones del proyecto MDD.
- **BP8. Requisitos para la validación de la cadena de herramientas.** Un desarrollador de la aplicación de negocios nunca deberá modificar un artefacto MDD ya generado; las herramientas, deberán estar totalmente integradas con el Sistema de Gestión de la Configuración.
- **BP9. Generación automática de los artefactos.** Deberá ser posible regenerar todos los artefactos de la aplicación de negocios en forma automática, a partir de un archivo generado para ese fin. De este modo, si fuera necesario ampliar en parte una transformación durante la construcción de la aplicación de negocios, todo puede regenerarse en forma automática.
- **BP10. Seguimiento y control.** Una vez construido el plan de proyecto, el seguimiento y control del proyecto MDD no es diferente al de cualquier otro proyecto de desarrollo de software.
- **BP11. El éxito.** El éxito de un proyecto MDD depende del éxito en la reutilización de los artefactos de los modelos. Esto incluye la identificación y recuperación de un artefacto

- para su reuso; asegurarse que se recupera el artefacto adecuado para la versión de ejecución que corresponde; chequear la integridad de un artefacto y verificar si la versión es la apropiada.
- **BP12. El seguimiento.** El seguimiento de un proyecto MDD es similar a cualquier otro proyecto de desarrollo de software. Pero hay ventajas adicionales que aporta el MDD, derivadas de su automatización.
 - **BP13. El ciclo de vida de un proyecto.** El marco de trabajo cubre la creación, testeo y desarrollo de los modelos, patrones y transformaciones que generarán la solución.
 - **BP14. Versiones.** Debe existir un mecanismo para el reemplazo, o el desarrollo de nuevas versiones que pueden coexistir, y asegurarse que sean accesibles por los usuarios adecuados.
 - **BP15. Nivel de versionado.** Debe determinarse el nivel de versionado (por archivo, por clase, por servicio, unidad de desarrollo y otros) a aplicar. Se versionan transformaciones, patrones, perfiles y todos los artefactos reusables.
 - **BP16. Certificación de servicio del modelo o artefacto.** Se recomienda tener un mecanismo para certificar que los artefactos y modelos cumplan los estándares y se mantenga la integridad del sistema.
 - **BP17. Depuración.** El código generado no debe ser depurado, deben depurarse los modelos y transformaciones, por dos razones: a) Es muy difícil volver atrás desde el código al problema subyacente en el modelo, b) Es crucial que todos los cambios se hagan en los modelos o transformaciones y no en los artefactos generados. Esta práctica asegura la consistencia de los modelos y la solución generada.
 - **BP18. Validación y testeo.** Deben validarse los artefactos solución contra los requerimientos de la solución y la lógica de negocios de los servicios. El testeo en MDD puede describirse en dos fases: a) del framework del modelo, b) de los artefactos solución generados
 - **BP19. Información válida.** Los modelos y transformaciones en un desarrollo MDD deben poblarse con información exacta y válida.
- Las siguientes prácticas fueron extraídas del libro de C. Pons, R. Giandini y G. Pérez [9]:
- **BP20. Expertos.** La plataforma MDD debe ser desarrollada por los profesionales más experimentados, que son:
 - Los expertos en el dominio quienes tienen conocimiento del dominio del problema y conocen la terminología y los conceptos y reglas del dominio.
 - Los desarrolladores del lenguaje quienes seleccionan y/o crean el lenguaje de modelado.
 - Los modeladores o ingenieros del PIM.
 - El ergonomista, quien ayuda a los desarrolladores a mejorar la usabilidad del lenguaje.
 - Los desarrolladores de las transformaciones y de los generadores de código especifican las transformaciones del PIM al PSM y del PSM al código.
 - Los expertos en el marco del dominio o ingenieros del PSM quienes son arquitectos de software o desarrolladores con experiencia en el uso y desarrollo de componentes y marcos de trabajo.
 - **BP21. Iteraciones.** Es aconsejable separar el desarrollo en varias iteraciones.
 - **BP22. Guías.** Se recomienda tener en cuenta las siguientes guías durante el desarrollo del proyecto: Realizar una inversión explícita en las herramientas de soporte. Utilizar a la gente más calificada para desarrollar las herramientas MDD con el objetivo de capturar y automatizar su experiencia. Considerar que además del código, el proyecto generará documentos, configuraciones, reportes y casos de prueba. Asegurarse que el proceso de desarrollo soporta ambientes de prueba además de ambientes de producción. Definir las estrategias de manejo de configuraciones para las herramientas MDD. Asignar tiempo al entrenamiento del equipo sobre el uso de herramientas MDD. Destinar tiempo para considerar si las herramientas MDD serán reusables en proyectos futuros.
 - **BP23. Métricas.** Al finalizar el proyecto MDD, es útil generar métricas: El costo de desarrollo de las herramientas MDD. La productividad de los desarrolladores de la aplicación al usar las herramientas comparando con el esfuerzo que hubiera sido necesario para desarrollar todo el código manualmente. El nivel de calidad logrado por el equipo de desarrollo; El esfuerzo requerido para lograr que las herramientas MDD puedan ser reutilizadas en otros proyectos.
 - **BP24. Herramientas.** Identificar, desarrollar e instalar las herramientas MDD requeridas, antes que los desarrolladores de la aplicación de negocios las necesiten.

- **BP25. Gestión.** La gestión de los artefactos MDD, sus descripciones relacionadas y el mantenimiento de sus repositorios se torna un tema relevante.

Las siguientes prácticas fueron extraídas del artículo de E. Ríos, T. Bozheva, A. Bediaga, N. Guilloreau. A. Rensink y J. Warmer [10].

Los autores en [10] introducen un modelo de madurez para la introducción de MDD en una organización. Este modelo consta de cinco niveles de capacidad. Los autores definen el nivel de madurez MDD de una organización de acuerdo a la evaluación de dos factores: si las prácticas y elementos MDD correspondientes al nivel existen o no existen. Y, por otro lado, si los atributos de los elementos MDD toman los valores apropiados que corresponden a dicho nivel de madurez. Para ello los autores consideran dos niveles. El nivel de madurez 1, corresponde a situaciones en que las prácticas de modelado en una organización, se utilizan en forma esporádica o directamente no se utilizan. Para el nivel de madurez 1 no se definen prácticas. El nivel de madurez 2, corresponde a organizaciones de mayor madurez en el modelado y en cada proyecto se crea un modelo técnico con el que el código final y la documentación del sistema deben ser coherentes. En este nivel, el modelo técnico combina los aspectos técnicos y de negocios del sistema a desarrollar sin distinguir entre ellos. Para el nivel de madurez 2, que los autores llaman *MDD Básico*, se definen las siguientes prácticas:

- **BP26. Técnicas de modelado.** Identificar las técnicas de modelado.
- **BP27. Modelo técnico.** Definir el modelo Técnico.
- **BP28. Generación de código.** Generar código a partir del modelo técnico.
- **BP29. Documentación.** Generar documentación a partir del modelo técnico.
- **BP30. Completar el código.** Completar el código para cumplir todos los requerimientos.
- **BP31. Selección de herramientas.** Decidir las herramientas de modelado.

El nivel de madurez 3, llamado MDD inicial, es aquel en que la organización comienza a desarrollar sistemas con un enfoque más cercano a MDD, en el que, además de alinear el código y los modelos, se

desarrollan modelos de negocios que direccionan la lógica de negocios en forma separada de la técnica. Los modelos de negocios se convierten en forma manual a los modelos técnicos, pero dichos modelos técnicos se representan por medio de una herramienta y se convierten automáticamente a código. Para el nivel de madurez 3 se definen las siguientes prácticas:

- **BP32. Modelo.** Definir el modelo de negocios.
- **BP33. Transformaciones.** Definir transformaciones para pasar del modelo al código.
- **BP34. Separación del código generado.** Separar el código generado del no generado.
- **BP35. Chequeo.** Chequear los modelos.
- **BP36. Workflow.** Definir el Workflow del proyecto MDD.
- **BP37. Cobertura.** Decidir la cobertura de las actividades de modelado.
- **BP38. Repositorios.** Establecer y mantener repositorios para los modelos y transformaciones.
- **BP39. Medidas.** Definir, recoger y analizar medidas con respecto a las actividades de modelado.

El nivel de madurez 4, denominado MDD Integrado, es aquel en el que la organización comienza a integrar sus modelos. Los modelos de negocios se derivan de los de dominio y se desarrollan por medio de una herramienta. Luego se transforman automáticamente a modelos técnicos y estos en código. Los conceptos de dominio, negocios y técnico están separados. Para el nivel de madurez 4 se definen las siguientes prácticas:

- **BP40. Metamodelo.** Definir el metamodelo centrado en la arquitectura.
- **BP41. Modelo de dominio.** Definir el modelo de dominio.
- **BP42. Transformaciones.** Definir las transformaciones del modelo de negocios al modelo técnico.
- **BP43. Simulación.** Simular modelos.
- **BP44. Separación.** Separar los modelos técnicos del producto e infraestructura de la familia de sistemas.
- **BP45. Administración de la infraestructura.** Administrar el desarrollo de la infraestructura común.

El nivel de madurez 5, llamado MDD Final, es aquel en el que las transformaciones entre modelos se

realizan en forma automática y además los modelos están completamente integrados entre ellos y con el código. En este caso, el foco del esfuerzo de la organización está en los modelos y no en el código. Si se alcanza este nivel de madurez, significa que el ciclo de vida completo del desarrollo está dirigido por modelos. Para el nivel de madurez 5, se definen las siguientes prácticas:

- **BP46. DSLs.** Definir lenguajes de dominio específicos.
- **BP47. Mejora y validación de metamodelos.** Mejorar y validar continuamente los metamodelos.
- **BP48. Transformaciones.** Definir transformaciones del modelo de dominio al modelo de negocios.
- **BP49. V&V.** Validación y verificación basadas en el modelo.
- **BP50. Elementos estratégicos.** Establecer y mantener los elementos MDD estratégicos.

ANÁLISIS DE LAS BUENAS PRÁCTICAS DE MDD EN EL CONTEXTO DE CMMI

A continuación, se discute para cada Área de Proceso cuáles son las prácticas de MDD que las soportan. Para ello, se busca que existan en MDD, actividades, artefactos, workflows, procedimientos o personas, que implementen las prácticas específicas de cada una de ellas. Para identificar cada práctica específica se utilizará la sigla SP (por su nombre en inglés “Specific Practice”) seguida por un número de la forma “x.y”. La “x”, es el número de objetivo específico al que corresponde la práctica específica. La “y”, es el número de secuencia de la práctica específica dentro de dicho objetivo. Esta nomenclatura, se utilizará en todo el trabajo para referirse a las prácticas específicas CMMI-DEV 1.3.

Se hace notar que no todas las buenas prácticas MDD seleccionadas, han sido utilizadas para evaluar el Nivel 2 de CMMI-DEV 1.3; sin embargo, la enunciación completa tiene por objeto dejar abierto el análisis para los demás niveles CMMI-DEV 1.3 que escapan al alcance de este trabajo. Para facilitar la comprensión del análisis completo que se presentará en la siguiente sección, se mostrará aquí como ejemplo la exploración detallada de dos prácticas específicas.

Ejemplo 1: En el Área de Proceso Gestión de la Configuración, la práctica específica SP1.1 expresa “Identificar los Ítemes de Configuración”. MDD brinda soporte a esta práctica mediante las siguientes prácticas: las prácticas BP1/BP6 identifican los artefactos MDD (o los ítemes de configuración) que deben generarse; la práctica BP24 indica que se deben identificar los artefactos MDD previo a que los desarrolladores de la aplicación de negocios los necesiten; las prácticas BP26/BP30, BP40/BP43, BP46 y BP48 indican los artefactos MDD/ítemes de configuración que deberán generarse durante el desarrollo. En conclusión, existen prácticas MDD que indican actividades, que al cumplirse satisfacen el objeto de la práctica CMMI.

Ejemplo 2: La práctica específica SP1.1 del Área de Proceso Gestión de los Requerimientos expresa “Comprender los Requerimientos”. Es soportada por las prácticas MDD BP1 a BP6, que apuntan a la comprensión de los requerimientos para luego construir el aplicativo de negocios. Las prácticas BP26, BP30, BP31, BP37 y BP39 indican los procedimientos a seguir para construir la cadena de herramientas MDD comprendiendo los requerimientos. Por otra parte, la práctica específica SP 1.5 de la misma área, define “Asegurar el Alineamiento entre los productos de trabajo y los Requerimientos”. En este caso, el soporte MDD se basa en la práctica MDD BP5, que indica que deben producirse artefactos de muestra para los escenarios clave, y la práctica BP6 que habla de la necesidad de validar la cadena de herramientas para garantizar el alineamiento de los productos de trabajo con los requerimientos. Además, la aplicación de la práctica MDD BP11, asegura que se mantendrá la trazabilidad y alineamiento entre los productos de trabajo y los requerimientos. Por lo tanto, existen prácticas MDD que indican actividades, que al cumplirse satisfacen el objeto de las práctica CMMI mencionadas.

Se describen a continuación los resultados obtenidos para cada una de las Áreas de Proceso Nivel 2 de CMMI-DEV 1.3 y que fueran, preliminarmente descritos en [11].

Área de Proceso Gestión de la Configuración

De acuerdo a CMMI-DEV [5], el propósito de esta Área de Proceso es el de establecer y mantener la integridad de los productos de trabajo utilizando la

identificación, el control, el status y las auditorías, de la configuración. Luego de analizar una por una, las prácticas específicas (SP) del área, se concluye que son abundantes las buenas prácticas (BP) MDD que aplican para dar soporte a esta Área de Proceso. La Tabla 1 muestra las BP que dan soporte a cada SP del área. Se concluye que, de las 7 prácticas específicas de CMMI-DEV 1.3, MDD soporta 7, por lo que el soporte es del 100%.

Área de Proceso Gestión de los Requerimientos

De acuerdo a CMMI, el propósito del Área de Proceso Gestión de los Requerimientos es administrar los requerimientos de los productos y componentes y asegurar el alineamiento entre dichos requerimientos y los planes de proyecto y productos de trabajo. En este caso, el soporte MDD es total, dado que hablar de requerimientos en un proyecto MDD, significa definir las características y gestión de los artefactos MDD, y los procedimientos para hacerlo, están detalladamente enunciados por todos los autores que tenemos como referencia. La Tabla 2 muestra las BP que dan soporte a cada SP del área.

Esta Área de Proceso posee 5 prácticas específicas, de las cuales MDD soporta todas, lo que significa el 100% del total.

Área de Proceso Aseguramiento de la Calidad del Proceso y del Producto

El propósito de esta Área de Proceso es el de proveer al staff y gerencia de una visión y comprensión

Tabla 1. Soporte MDD para cada práctica específica del Área Gestión de la Configuración.

SP	Definición de la SP	BPs que la soportan
1.1	Identificar los ítemes de Configuración	BP1-6, BP24, BP26-30, BP40-43, BP46, BP48.
1.2	Establecer un sistema de Gestión de Configuración	BP22, BP25, BP8, BP9, BP11, BP14, BP38, BP50.
1.3	Crear líneas base	BP1-6, BP27-33, BP36, BP40-42.
2.1	Rastrear los pedidos de cambio	BP9.
2.2	Controlar los ítemes de Configuración	BP9, BP17, BP18.
3.1	Establecer registros de Gestión de Configuración	BP8, BP9 y BP50.
3.2	Realizar auditorías de Configuración	BP8.

Tabla 2. Soporte MDD para cada práctica específica del Área Gestión de los Requerimientos.

SP	Definición de la SP	BPs que la soportan
1.1	Comprender los Requerimientos	BP3, BP5, BP13 y BP20.
1.2	Obtener compromiso con los Requerimientos	BP6, BP13 y BP22.
1.3	Administrar los cambios de Requerimientos	BP14 y BP15.
1.4	Mantener la trazabilidad bidireccional de los Requerimientos	BP11, BP15, BP16, BP18, BP25, BP49 y BP50.
1.5	Asegurar el alineamiento entre los productos de trabajo y los Requerimientos	BP5, BP6 y BP49.

objetiva de los procesos y productos de trabajo asociados. Se ha verificado un alto soporte MDD a esta Área de Proceso como puede observarse en los datos desplegados en la Tabla 3. Esta área de Proceso posee 4 prácticas específicas, de las que MDD soporta 3, que corresponden al 75% del total.

Áreas restantes

Por limitaciones de espacio no se ha incluido en este artículo el análisis detallado de las demás Áreas. El lector interesado puede consultarlo en [12]. La Tabla 4 resume los resultados obtenidos.

ANÁLISIS DE LAS FALENCIAS EN EL SOPORTE QUE BRINDA MDD A CADA ÁREA DEL CMMI-DEV

En esta sección se discuten las posibles causas y consecuencias de las falencias en el soporte que MDD brinda a cada área del CMMI-DEV.

– **Área de Proceso Gestión de la Configuración:** Esta Área de Proceso tiene alto soporte MDD: el 100%, o sea el máximo posible.

– **Área de Proceso Gestión de los Requerimientos:** Esta Área de Proceso tiene alto soporte MDD: el 100%, o sea el máximo posible.

– **Área de Proceso Aseguramiento de la Calidad del Proceso y del Producto:** Esta Área de Proceso también tiene alto soporte MDD: el 75%. Solo hay una práctica específica CMMI-DEV 1.3, que no tiene soporte y es la SP 2.1 que significa “Comunicar y

solucionar problemas no resueltos”. El texto CMMI, expresa, para la práctica específica SP 2.1, que “Los problemas de incumplimiento son problemas identificados en evaluaciones que reflejan falta de adherencia a estándares aplicables, descripciones de procesos o procedimientos. El status de los problemas de incumplimiento indica las tendencias en la calidad”.

Tabla 3. Soporte MDD para cada práctica específica del Área Aseguramiento de la Calidad del Proceso y del Producto.

SP	Definición de la SP	BPs que la soportan
1.1	Evaluar objetivamente los procesos	MDD BP6, BP8, BP13, BP16, BP22, BP35 y BP50.
1.2	Evaluar objetivamente los productos de trabajo	BP5, BP11, BP16, BP18, BP22 y BP25.
2.1	Establecer registros	BP11, BP16 y BP50.
2.2	Comunicar y solucionar los problemas no resueltos	no es soportada.

Tabla 4. Soporte MDD para cada Área de Proceso CMMI.

Área de Proceso	Nº Total de SPs	Nº de SPs soportadas por MDD	% soportado por MDD
Gestión de la Configuración	7	7	100
Gestión de los Acuerdos con Proveedores	6	0	0
Gestión de los Requerimientos	5	5	100
Aseguramiento de la Calidad del Proceso y del Producto	4	3	75
Medición y Análisis	8	3	37,5
Monitoreo y Control del Proyecto	10	5	50
Planeamiento del Proyecto	14	12	86

Los ejemplos de productos de trabajo que se indican en CMMI son: Reportes de acciones correctivas, Reportes de evaluación y Tendencias de calidad. El problema en MDD reside en la falta de procedimientos que indiquen la necesidad de registrar los problemas de incumplimiento. Por ejemplo la práctica MDD BP5. Producir los

Artefactos de Muestra para los escenarios clave, aplica al control de los problemas no resueltos, pero no incluye la necesidad de generar procedimientos para su registro. En este caso, la práctica BP5 debería tener una sub-práctica que exprese la necesidad de que, en caso de detectar fallas en los artefactos de muestra, se confeccionen Reportes de acciones correctivas y Reportes de evaluación.

– **Área de Proceso Medición y Análisis:** Constituye el Área de Proceso de Nivel 2 con más bajo soporte MDD: 37,5 % de soporte. Las prácticas específicas CMMI no soportadas son:

- SP1.4. Especificar procedimientos de Análisis.
- SP2.1. Obtener Datos de Mediciones.
- SP2.2. Analizar Datos de Mediciones.
- SP2.3. Almacenar Datos y Resultados.
- SP2.4. Comunicar los Resultados,

Vemos que, de las 4 prácticas específicas del objetivo específico 1 que significa “Alinear las actividades de medida y análisis”, están soportadas 3. Ninguna de las prácticas específicas del objetivo específico 2 está soportada. En general, si analizamos el soporte MDD a las prácticas específicas del objetivo específico 1, se observa que el soporte encontrado, aunque existe, está basado en pocas prácticas MDD sobre todo en lo que hace a las prácticas específicas SP 1.1 y SP 1.2 que se refieren a la necesidad de efectuar mediciones, para las que únicamente se encontró la práctica MDD BP23 referente a la utilidad de generar métricas a posteriori del proyecto MDD. Analicemos ahora, la práctica específica SP1.4 “Especificar Procedimientos de Análisis”, del objetivo específico 1, que no es soportada por MDD. La práctica se orienta a la especificación de procedimientos de análisis que permitan detallar cómo se analizan y comunican los datos medidos. De acuerdo al CMMI- DEV 1.3 se entiende por datos a la información grabada que puede incluir datos técnicos, documentos de software de computación, información financiera, representación de hechos, números o datos de cualquier naturaleza que puedan ser comunicados, almacenados y procesados. En el caso de MDD, los datos posibles serían datos técnicos, documentos de software de computación y otros datos que hacen al proyecto MDD. Por lo que, para soportar la práctica específica, deberían especificarse cuáles serían los procedimientos adecuados para el análisis de las herramientas

MDD. En términos generales, se verifica que si bien están indicadas las acciones de generación de dichas herramientas, que son los datos de este caso, las prácticas MDD en general no indican los procedimientos para registrarlos y analizarlos.

Sobre el objetivo específico 2, que establece “Proveer Resultados de las Mediciones”, y que corresponde a las prácticas específicas no soportadas, SP2.1, SP2.2, SP2.3 y SP 2.4, se explica en CMMI, que, la razón primaria para conducir la medición y el análisis es satisfacer las necesidades identificadas de información, derivadas de los objetivos organizacionales y de negocios. En este caso, las prácticas específicas CMMI se refieren a la necesidad de obtener, registrar y almacenar los resultados de las mediciones para obtener información. El registro de los resultados obtenidos, tema similar al analizado para la práctica específica SP 1.4 analizada antes, es un punto débil de MDD que también se señala para el Área de Proceso Aseguramiento de la Calidad del Proceso y del Producto, que debiera ser resuelto con recomendaciones de los especialistas que dieran lugar a nuevas prácticas MDD.

– **Área de Proceso Monitoreo y Control del Proyecto:** Esta es la segunda Área de Proceso de menor soporte MDD (50% de soporte). Las prácticas específicas no soportadas son:

- SP1.6 Conducir las Revisiones del Progreso
- SP1.7 Conducir las Revisiones de los Hitos
- SP2.1 Analizar los Problemas
- SP2.2 Tomar Acciones Correctivas
- SP2.3 Administrar las Acciones Correctivas

En forma general, el insuficiente soporte de MDD a este Área de Proceso, se debe en gran parte, en nuestra opinión, a que los autores en los que se ha basado este trabajo para la selección de buenas prácticas, afirman, que el seguimiento de un proyecto MDD es similar al de cualquier otro proyecto de software y no se han fijado prácticas ni recomendaciones que apunten a la problemática especial del seguimiento y control del proyecto MDD. Sin embargo, hay cuestiones que podrían ser analizadas en particular para un proyecto MDD, como indicar cuáles son los problemas específicos del MDD que deberían analizarse, cuáles deberían ser las acciones correctivas a lo largo del desarrollo, cuáles deberían ser los hitos clave y cómo deben administrarse las acciones correctivas entre

otros. Además, en el área que analizamos, puede verse en el análisis hecho anteriormente, que aún para las prácticas específicas soportadas se han encontrado pocas prácticas MDD. Estudiemos, con un ejemplo, cómo mejorar el soporte para prácticas específicas que tienen soporte limitado como es el caso de la práctica específica SP 1.5 Monitorear la participación de los stakeholders soportada por la práctica MDD BP24 únicamente. En MDD, los stakeholders más significativos son los desarrolladores de la aplicación de negocios que no participan en el proyecto MDD pero serán sus usuarios. Deberían generarse, además de la práctica BP24 para soportarla, otras que garanticen la participación de los stakeholders a lo largo de todo el ciclo de vida del proceso MDD.

– **Área de Proceso Planeamiento del Proyecto:** Este Área de Proceso tiene un alto grado de soporte MDD (86%). Las únicas dos prácticas para las que no se ha encontrado soporte MDD son:

- SP3.1 Revisar los planes que afectan el proyecto
- SP3.2 Reconciliar niveles de trabajo y recursos

A pesar del alto soporte para esta área encontrada, ha sido la que obtuvo resultados más bajos en la validación de profesionales, sobre todo en lo que se refiere a estimar costo y esfuerzo en el proyecto, referido a la práctica específica SP 1.4. Pero, si se elimina el soporte MDD propuesto en este trabajo a dicha práctica específica se obtendría un soporte MDD de 78,57% para el Área de Proceso Planeamiento del Proyecto, que sigue siendo alto. Por lo que la consulta no invalida los resultados obtenidos, pero el tema costos y esfuerzo merece ser discutido especialmente. La práctica MDD BP6, indica que luego de definir la cadena de herramientas “es posible crear un plan detallado del esfuerzo necesario para construir las herramientas MDD”, pero, se dice “posible” y no “obligatorio” y además no está enunciado como una tarea independiente que entendemos sería lo adecuado. La recomendación sería incluirlo como tarea explícitamente en la lista de tareas a realizar en el proyecto MDD. De hecho, se entiende que no se le está otorgando la suficiente importancia a la tarea de estimar el esfuerzo y costo de un proyecto MDD. Debe tenerse en cuenta que la poca claridad sobre los costos del proyecto es uno de los impedimentos más importantes para la adopción de MDD de los gerentes y directores de sistemas en las organizaciones. Si bien se expresa en forma

general que utilizar MDD puede ahorrar costos, la realidad es que al comenzar su implementación en una empresa esto no es así y no está claro si en la práctica será así en el futuro. Por eso es que debería analizarse cómo los costos se van mitigando con el tiempo a medida que se construyen los repositorios y el reuso se hace factible. Además, si bien existen prácticas MDD recomendando la evaluación del reuso de los artefactos al momento de ser diseñados y construidos (prácticas BP11, BP22 y BP38), existen pocas prácticas concretas que aseguren que esto se haga efectivo. Por ejemplo, la práctica BP2 aplica al reuso en sí misma al indicar que es necesario “asegurarse que en los proyectos siguientes se incluirán tareas que encaren el reuso”, pero no se enumeran las tareas que es necesario realizar para alcanzar este objetivo. Otro punto a mencionar, es en relación a la práctica específica SP 1.4 “Estimar el Esfuerzo y Costo”, para la que se han encontrado solo dos prácticas MDD que la soportan, que son BP6 y BP23. Debido a el documento CMMI-DEV 1.3, cuando se analiza esta práctica específica en el Área de Proceso Planeamiento del Proyecto se dice que “las estimaciones de esfuerzo y costo están generalmente basadas en resultados de análisis utilizando modelos de datos históricos” esto es razonable en el caso de MDD, dado que aún no existen suficientes datos históricos y se torna dificultoso hacer dichas estimaciones. Este punto débil del MDD se debe a que todavía no existe suficiente experiencia de uso y reuso de los artefactos MDD en las organizaciones. Pero debería evaluarse como debieran evolucionar los costos, idealmente y bajo qué condiciones, a lo largo del tiempo, en la medida en que una organización gane experiencia y construya su repositorio.

Análisis global

Puede apreciarse que algunas prácticas específicas CMMI están soportadas por 2 prácticas MDD únicamente mientras que otras, particularmente la SP 2.1 del Área de Planeamiento del Proyecto, tiene 34 prácticas MDD que la soportan. Por eso, parece razonable fijar un rango a partir del que pueda hablarse de un soporte MDD débil o ausente y soporte MDD fuerte. Tomando los valores promedio de número de prácticas específicas soportadas por MDD, en una Área de Proceso determinada, es razonable tomar un límite de 2 prácticas MDD que es el valor que corresponde a las dos Áreas de Proceso con menor soporte MDD, que son Monitoreo

y Control del Proyecto y Medición y Análisis. Teniendo en cuenta los resultados obtenidos en este trabajo, parece razonable decir, que si el número promedio de prácticas específicas soportadas por MDD, es menor o igual a 2 el soporte es débil y en caso contrario el soporte es fuerte. El resumen de esta definición aparece en la Tabla 5.

Tabla 5. Número promedio de prácticas específicas CMMI soportado por prácticas MDD por Área de Proceso.

Área de Proceso	Soporte promedio (#BPs/#SPs)	Tipo de soporte
Gestión de la Configuración	7,28	Fuerte
Gestión de los Requerimientos	3,8	Fuerte
Aseguramiento de la Calidad	5,33	Fuerte
Medición y Análisis	2	Débil
Monitoreo y Control	2	Débil
Planeamiento del Proyecto	11,16	Fuerte

VALIDACIÓN DE LA PROPUESTA

Para validar el análisis realizado en el presente trabajo, en relación al soporte MDD del Nivel 2 del CMMI-DEV 1.3, se procedió a recabar la opinión de otros profesionales sobre los resultados y conclusiones obtenidos. Para ello se elaboró una planilla de encuesta donde los profesionales podían manifestarse a favor o en contra de cada ítem de la propuesta. Los perfiles consultados corresponden a cinco profesionales de ingeniería de software especialistas en Gestión de Calidad. La consulta se realizó sobre 3 Áreas de Proceso de Nivel 2, y dentro de cada una, sobre dos de sus prácticas específicas. Cada una de las áreas encuestadas fue respondida por tres profesionales. Las tres áreas consultadas fueron: Área de Proceso Gestión de la Configuración, Área de Proceso Gestión de los Requerimientos, y Área de Proceso Planeamiento del Proyecto. La Tabla 6 muestra el soporte obtenido para las dos prácticas específicas encuestadas del Área de Proceso Gestión de las Configuraciones. Cada columna muestra el porcentaje de aprobación expresado por cada profesional encuestado, mientras que la última columna muestra el valor promedio obtenido para cada SP. La Tabla 7 muestra el soporte obtenido para las dos prácticas específicas encuestadas del Área de Proceso Gestión de los Requerimientos.

El soporte MDD para estas prácticas específicas queda corroborado con el resultado de la consulta.

Tabla 6. Resultado de la Encuesta del Área de Proceso Gestión de la Configuración.

SP	Prof 1	Prof 2	Prof 3	Prom.
SP 1.1	87,5%	100%	100%	95,8%
SP 2.1	100%	100%	100%	100%

Tabla 7. Resultados de la encuesta del Área de Proceso Gestión de los Requerimientos.

SP	Prof 1	Prof 2	Prof 3	Prom.
SP 1.1	100%	40%	100%	80%
SP 1.5	100%	100%	100%	100%

La Tabla 8 muestra el soporte obtenido para las dos prácticas específicas encuestadas del Área de Proceso Planeamiento del Proyecto.

Tabla 8. Resultados de la encuesta del Área de Proceso Planeamiento del Proyecto.

SP	Prof 1	Prof 2	Prof 3	Prom.
SP 1.1	85,7%	80,9%	38,1%	65,2%
SP 1.4	50%	50%	100%	66,6%

Los porcentajes de aprobación de los profesionales consultados, son menores para esta Área de Proceso comparados con las analizadas con anterioridad, pero queda validado el soporte MDD para la práctica SP1.1 y se obtiene un soporte débil para la práctica específica SP 1.4, dadas las razones que se pasa a enumerar. En el caso de la práctica específica SP 1.1 que significa “Establecer una estructura WBS (Work Breakdown Structure) de alto nivel para estimar el alcance del proyecto”, el porcentaje promedio es del 65,23% debido sobre todo a la baja evaluación del profesional 3. Esto no invalida el soporte MDD a esta práctica específica ya que dicho profesional ha calificado con la respuesta “sí”, a 8 de las 21 prácticas específicas MDD propuestas. Al analizar cuáles son las prácticas MDD aprobadas por el profesional 3, se concluye que las tareas indicadas en cada una de ellas permitirían generar una estructura WBS adecuada para estimar el alcance del proyecto. En

relación a la práctica específica SP 1.4, se obtuvo acuerdo de los tres consultados con el soporte de la práctica BP6 que significa “Definir la cadena de herramientas MDD”. Pero dos profesionales objetaron el soporte brindado por la práctica BP23. Consideramos que la consulta para la práctica SP 1.4 del Área de Proceso Planeamiento del Proyecto resulta parcialmente satisfactoria ya que son válidas las objeciones planteadas. Entendemos que dadas las respuestas obtenidas para soportar la SP 1.4 debe considerarse como soporte MDD, únicamente la práctica BP6. Esto lleva a un resultado final, para las tres Áreas de Proceso, que se expresa en la Tabla 9.

Tabla 9. Resultados del Proceso de Validación sobre soporte MDD a las tres áreas consultadas.

Área de Proceso	Soporte MDD (Resultado de Validación)
Gestión de la Config.	Sí, para las dos prácticas específicas consultadas
Gestión de los Req.	Sí, para las dos prácticas específicas consultadas
Planeamiento del Proyecto	Sí, para SP 1.1 Sí débil, para SP 1.4

Los detalles del proceso de encuestas se encuentran en el documento de Tesis de V. Esterkin [12].

TRABAJOS RELACIONADOS

Iñigo Garro, consultor y colaborador del SEI en el Quality Working Group, constituido para mejorar la calidad de las evaluaciones CMMI y las prácticas de la profesión, transcribe en [13], basándose en el informe técnico CMU/SEI-2010-TR-033 del Software Engineering, las directrices del modelo CMMI relacionadas con las metodologías ágiles, y además, explica aspectos claves que permiten la coexistencia de CMMI y los enfoques ágiles. Si bien este trabajo se enfoca sobre otro paradigma de desarrollo de software, es conveniente tenerlo en cuenta ya que brinda las bases para la definición del mapeo entre el CMMI y los distintos paradigmas de desarrollo.

En [10] los autores muestran cómo un modelo de madurez desarrollado dentro del proyecto “Modelware” ayudó a varias compañías a adoptar

el modelo de desarrollo MDD. El proyecto fue desarrollado en España durante los años 2002 al 2006 y definía cinco niveles de madurez que ubicaban a las empresas que adoptaban MDD en diferentes grados, tanto de seguimiento de prácticas MDD como de uso de artefactos MDD. En lugar de definir diferentes niveles de cumplimiento de prácticas MDD, este artículo propone adoptar un modelo de calidad reconocido, como es el CMMI, respondiendo a la necesidad de integrar el control de las prácticas específicas de desarrollo en este nuevo paradigma de desarrollo de software.

El estudio realizado por J. Quintero, P. Rucinke, R. Anaya y G. Piedrahita [14] enuncia los problemas “top” de MDA y presenta recomendaciones sobre cómo encararlos y mitigarlos. En el presente trabajo se ha investigado si, al aplicar dichas recomendaciones, puede mejorarse el soporte MDD. Se ha determinado, que el análisis de las prácticas específicas CMMI Nivel 2, a la luz de los problemas top de MDD, y las recomendaciones para encararlos, introducen pocos elementos nuevos en cuanto a las prácticas específicas no soportadas por MDD, pero en cambio refuerzan el soporte con nuevos elementos en algunas prácticas específicas que sí soporta MDD. Esto es razonable dado que el Nivel 2 de CMMI es el nivel gerenciado y el artículo de Quintero *et al.*, analiza las herramientas desde el ángulo técnico, para “lograr una mejor comprensión del modo de encarar los 10 problemas “top” de adopción desde el punto de vista de los desarrolladores”. Al analizar los 10 problemas se observa que los nueve primeros, son cuestiones técnicas referidas a los modelos y las herramientas de modelado utilizadas para generarlos, y uno se refiere a la cultura organizacional y su agrado del uso de los modelos. Sin embargo, en algunos casos, sus recomendaciones técnicas podrían ayudar a incrementar el soporte MDD al Nivel 2 de CMMI al incorporar a su análisis las herramientas utilizadas para el desarrollo MDD, sus características y costos. Es el caso de la práctica SP 1.4 “Estimar el Esfuerzo y Costo”, del Área de Planeamiento del Proyecto. La recomendación para encarar el problema 7 “Las herramientas de modelado son muy costosas” que tiene como respuesta: “En este caso hay muchas herramientas de software libre, pero si en el proceso de selección la elegida es muy costosa, los primeros proyectos deben mostrar ganancias en costo, tiempo y calidad para justificar la inversión”, muestra un elemento adicional no considerado en las prácticas

MDD seleccionadas, que es el costo en sí mismo de las herramientas MDD que debe tenerse en cuenta para estimar el costo de un proyecto MDD. También se refuerza el soporte MDD a las prácticas específicas SP 1.3, SP1.4 y SP 1.5 del Área de Proceso Gestión de los Requerimientos con las recomendaciones para encarar los problemas 1, 9 y 5, respectivamente.

En [15], J. Quintero y R. Anaya plantean los retos que aún afronta el MDD, como son las limitaciones de las herramientas y la falta de integración de los modelo BPM en las transformaciones de modelos. En el artículo propuesto se hace también un reconocimiento de los riesgos actuales con ambas falencias, pero se analiza y se plantea una forma de mitigar estos riesgos mediante controles presentados en las notas que se sugiere incluir en el modelo CMMI. Nuestro trabajo agrega un análisis detallado de las prácticas del CMMI y su conexión con MDD.

A. Lins de Vasconcelos, G. Giachetti, B. Marín y O. Pastor, en [16] han definido un proceso de desarrollo de software basado en MDD, que va desde los requerimientos hasta el código final, que integra elementos del framework i* y la metodología GORE (Goal-Oriented Requirement Engineering) y que es compatible con CMMI-DEV. En este caso, los autores se han enfocado en la definición de un nuevo proceso de desarrollo de software, garantizando, por construcción, su compatibilidad con el modelo de madurez. Nuestro trabajo no se enfoca en ningún proceso en particular, sino en la metodología MDD en general y, por lo tanto, pretende ser aplicable a cualquier proceso de desarrollo MDD.

CONCLUSIONES

Hasta el momento MDD se focaliza en el trabajo técnico y en este sentido se puede concluir que existen muchas recomendaciones sobre la metodología y secuencias necesarias para el desarrollo de un proyecto MDD, como se muestra en el alto grado de soporte obtenido para las Áreas de Proceso, Gestión de la Configuración, Gestión de los Requerimientos, Aseguramiento de la Calidad del Proceso y del Producto y Planeamiento del Proyecto. Si bien, las Áreas de Proceso y prácticas específicas no soportadas, o débilmente soportadas por MDD se explican, en gran parte, por la reciente irrupción de MDD en el desarrollo de software, el trabajo actual

apunta a los problemas que deberían ir resolviéndose para estimular su uso en las organizaciones. En términos generales, se puede concluir que en MDD falta detallar cuestiones concretas como procedimientos, documentación, métodos de seguimiento y otras, como se ha señalado al analizar el soporte MDD a cada Área de Proceso. Todavía, los problemas de falta de soporte detectados no están resueltos en las Áreas de Proceso con menor soporte MDD como las de Medición y Análisis y Monitoreo y Control del Proyecto, y deberían encararse para mejorar el soporte MDD al CMMI Nivel 2.

Como propuesta para encarar en el futuro, sería interesante recoger en una investigación entre profesionales especialistas en ingeniería de software, sus propuestas y recomendaciones de prácticas MDD, que ayuden a la mejora del soporte al Nivel 2 del CMMI, ampliando el alcance de la encuesta de validación realizada para este trabajo. Por otra parte, para mejorar el soporte MDD al CMMI Nivel 2, se ha planteado la conveniencia de complementar el desarrollo MDD con el uso de la metodología RUP, como se describe en el artículo de P. Eeles [17] y L. Balmelli, D. Brown, M. Cantor y M. Mott [18]. Otro trabajo futuro consiste en evaluar dicha convivencia, apuntando a resolver algunas prácticas específicas no soportadas, sobre todo del Área de Proceso Monitoreo y Control del Proyecto.

REFERENCIAS

- [1] M. Brambilla, J. Cabot and M. Wimmer. "Model-Driven Software Engineering in Practice". Morgan&Claypool Publishers ISBN: 9781608458820. 2012.
- [2] T. Stahl and M. Voelter. "Model-Driven Software Development". Technology, Engineering, Management. 1st edition. Wiley. 2006.
- [3] D. Di Ruscio, R.F. Paige and A. Pierantonio, Editors. Science of Computer Programming. Special issue on Success Stories in Model Driven Engineering. Vol. 89, pp. 69-222. Copyright © 2014. Elsevier. 2014.
- [4] OMG success stories. En línea: http://www.omg.org/mda/products_success.htm, accedido en 2015. 2015.
- [5] CMMI-DEV 1.3. URL: <http://www.sei.cmu.edu/reports/10tr033.pdf>) liberada en 2010. 2010.
- [6] A. Kleppe, J. Warner and W. Bast. "MDA Explained, The Model Driven Architecture: Practice and Promise. Publisher: Addison-Wesley; 1st edition. ISBN: 032119442X. 2003.
- [7] Object Management Group. URL: www.omg.org
- [8] P. Swithinbank, M. Chessell, T. Gardner, C. Griffin, J. Man, H. Wylie and L. Yusuf. "Patterns: Model-Driven Development Using IBM Rational Software Architect". URL: <http://http://www.redbooks.ibm.com/redbooks>. 2005.
- [9] C. Pons, R. Giardini y G. Pérez. "Desarrollo dirigido por modelos". Editorial Edulp con Mc Graw-Hill Educación, pp. 279. ISBN: 978-950-34-0630-4. URL: http://www.editorial.unlp.edu.ar/libro_pons.html. 2010.
- [10] E. Ríos, T. Bozheva, A. Bediaga, N. Guilloreau. A. Rensink and J. Warmer. "MDD Maturity Model: A Roadmap for Introducing Model-Driven Development". A. Rensink and J. Warmer (Eds). ECMDA-FA 2006. LNCS 4066, pp. 78-89. 2006.
- [11] V. Esterkin and C. Pons. "Análisis y Evaluación del MDD. Model Driven Development desde la Perspectiva del Nivel 2 del CMMI DEV 1.3". Publicado en Proceedings of CACIC. Congreso Argentino de Ciencias de la Computación. Argentina. 2012.
- [12] V. Esterkin. "Análisis y Evaluación del MDD (Model Driven Development) desde la perspectiva de CMMI DEV 1.3 Nivel 2". Tesis para optar al grado de Magíster. Universidad Abierta Interamericana. Ciudad de Buenos Aires, Argentina. 2014.
- [13] I. Garro. "Interpretación de CMMI® para Desarrollo, Versión 1.3 en enfoques ágiles". 2013. URL: <https://inigogarro.files.wordpress.com/2013/12/interpretacion-de-cmmi-dev-v1-3-en-enfoques-c3a1giles-web1.pdf>
- [14] J. Quintero, P. Rucinke, R. Anaya and G. Piedrahita. "How face the top MDE adoption problems, An Exploratory Study Case (CLEI)". XXXVIII Conferencia Latinoamericana, pp. 1-10. Octubre 2012, Medellín. ISBN: 978-1-4673-0794. 2012.
- [15] J. Quintero y R. Anaya. "Marco de Referencia para la evaluación de herramientas basadas en MDD". 10° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software IDEAS'2007, pp. 225-238. 2007.

- [16] A. Lins de Vasconcelos, G. Giachetti, B. Marín and O. Pastor. "Towards a CMMI-Compliant Goal-Oriented Software Process through Model-Driven Development. "The Practice of Enterprise Modeling", pp. 253-267. ISBN: 978-3-642-24848-1. Publisher Springer Berlin Heidelberg. 2011.
- [17] P. Eeles. "MDA and RUP". IBM Software Group. IBM Corporation. 2004. URL: [http://www.architecting.co.uk/presentations/MDA and RUP.pdf](http://www.architecting.co.uk/presentations/MDA_and_RUP.pdf)
- [18] L. Balmelli, D. Brown, M. Cantor and M. Mott. "Model-Driven Systems Development". IBM Systems Journal - Model-Driven Software Development. Vol. 45, Issue 3. 2006.