CONTROL DE RIESGOS EN LA GENERACION DE APLICACIONES, UTILIZANDO EL DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS (MDSD). Revisión de la literatura.

John Deiby Salazar

Facultad de Ingenierías, Universidad de Medellín

{ *jdeissar@gmx.com* }

Resumen

El Desarrollo de Software Dirigido por Modelos tiene como principal objetivo especificar y explicitar los términos del negocio en modelos. Los modelos no solo al inicio del proceso de desarrollo de software, sino en todo el ciclo de vida a través de transformaciones. Con las transformaciones se le está ofreciendo a los desarrolladores la posibilidad de poder realizar la automatización de sus procesos llevando a un nivel de abstracción mayor la obtención de los artefactos. Utilizando los modelos, se mitigan los riesgos en los atributos de calidad del software, generados por la creciente complejidad de las aplicaciones, que han de satisfacer un mayor número de requisitos como: distribución, adaptabilidad, mantenibilidad y reutilización que hacen frente a la complejidad de las plataformas actuales. El presente artículo hace una revisión de diferentes artículos en los cuales se aborda el tema del uso del Desarrollo de Dirigido por Modelos para el control de riesgos en la generación de aplicaciones.

Palabras Claves: Control de riesgos, Calidad del Sofware, Desarrollo de Software Dirigido por Modelos, Model Driven Architecture (MDA), artefactos, transformaciones.

1. Introducción

La construcción tradicional de software se ha caracterizado por el uso intensivo de plataformas específicas y lenguajes de programación, generando una alta dependencia tecnológica, y con ello, el riesgo de descartar consideraciones de negocio esenciales para el correcto funcionamiento del software, y su posterior acoplamiento en el contexto de múltiples soluciones computacionales como soportes a los procesos de negocio críticos de cualquier compañía.

Una tendencia en la construcción de software es la aplicación del enfoque de desarrollo dirigido por modelos (Model-driven Development), donde los principales activos del proceso son los modelos elaborados a partir de las especificaciones claves del software, desde el punto de vista organizacional, funcional, estructural, de comportamiento, y otros atributos que inciden directamente en la calidad del producto, como la usabilidad (relación eficiente humano-computador). Mediante el enfoque MDD se busca la generación automática del código software como consecuencia misma de los modelos, en lugar de elaborar especificaciones cognitivamente complejas directamente sobre una tecnología en particular. El enfoque MDD prioriza el trabajo sobre altos niveles de abstracción, en donde los modelos permiten identificar y proponer soluciones a complejos procesos de negocios antes de llegar a plataformas tecnológicas concretas.

Hay varias razones que han motivado la aparición de este nuevo paradigma. Tenemos en primer lugar la creciente complejidad de las aplicaciones de software, que han de satisfacer un mayor número de requisitos(distribución, heterogeneidad, disponibilidad, adaptabilidad, etc.) con mejores prestaciones y menos tiempos de desarrollo, por otro lado sabemos que las nuevas tecnologías evolucionan muy rápido lo que hace que las inversiones en las tecnologías concretas sean demasiado volátiles. Si bien es cierto que esos problemas no son nuevos en el campo de la Ingeniería del software, está comprobado que la mejor forma de tratar con ellos es elevando el nivel de la abstracción de los modelos desde las primeras etapas del desarrollo.

Durante décadas se ha perseguido la meta de encontrar procesos reproducibles y predecibles que mejoren la productividad y la calidad. Algunas de estas soluciones

intentan sistematizar o formalizar la aparentemente desorganizada tarea de desarrollar software. Sin el desarrollo de software por modelos, los proyectos de software corren el riesgo de demorarse o consumir un presupuesto mayor que el planeado. Dada la cantidad de proyectos de software que no cumplen sus metas en términos de calidad, funcionalidad, costes o tiempo de entrega.

La revisión que se presenta sobre el control de riesgos en la generación de aplicaciones, utilizando un contexto que se está produciendo en un nuevo movimiento en el ámbito del desarrollo software denominado “dirigido por modelos” o Model Driven Development (MDD). Lo que se plantea en este movimiento es elevar un peldaño el nivel de abstracción con respecto a cómo se desarrolla software en la actualidad. Para conseguir el nuevo nivel de abstracción, este movimiento plantea que los modelos software sean artefactos de desarrollo de primer nivel. Estos modelos serán precisos, completos y no ambiguos, de tal modo que su traducción a código fuente ejecutable, pueda ser total o parcialmente realizada de forma automática. Del mismo modo que un compilador puede traducir de forma automática código fuente escrito en un lenguaje de programación de alto nivel a código máquina entendible por un ordenador, sería posible disponer de herramientas que transformasen modelos formales a código fuente de forma automática. De materializarse esta propuesta, haría que el modo en el que desarrollamos software hoy fuese visto como un proceso en extremo rudimentario, del mismo modo que hoy consideramos la manera en la que se desarrollaba software hace 4 décadas una disciplina primitiva.

Dentro del movimiento del desarrollo de software dirigido por modelos, ha surgido con fuerza una iniciativa denominada Model Driven Architecture (MDA). La razón es que tiene detrás un consorcio como el Object Management Group (OMG) y está edificada sobre un conjunto de estándares, entre los que destaca UML. Esta propuesta ha generado fuertes reacciones en la comunidad, tanto positivas por lo ambicioso de sus objetivos, como negativas y extremadamente escépticas, por la inexistencia en la actualidad de tecnologías y especificaciones lo suficientemente maduras como para construir y transformar en código modelos software precisos,

formales y de propósito general.

El desarrollo de este artículo se revisará con la aproximación de la situación ya descrita.

2. Método

Para llevar a cabo esta investigación sobre los riesgos en la generación de aplicaciones, se tuvo en cuenta para delimitar la problemática, el estudio de riegos de calidad del software, donde se analizaron diferentes perspectivas respecto a la temática de las transformaciones de modelo y se definió que el enfoque sería el control de riesgos en la generación de aplicaciones utilizando modelos. En el estudio de este tema, se describen hallazgos sobre requisitos de sistemas complejos. A partir de referencias bibliográficas de trabajos encontrados que soportan el contenido del artículo, se pudo realizar un despliegue de investigación y búsqueda avanzada. Detallando así, los métodos que fueron utilizados para obtener resultados acorde con el tema de MDSD y que ayudaron a encontrar un caso de estudio, teniendo en cuenta la lectura del resumen y las conclusiones de cada uno.

2.1 Preguntas de investigación

Esta propone analizar los siguientes aspectos referidos a las metodologías existentes para crear software, en tal sentido y a fin de determinar la idoneidad de las metodologías existentes, planteamos los siguientes interrogantes:

¿Cómo mitigar los riesgos de calidad en el desarrollo de aplicaciones, utilizando la metodología de desarrollo de software dirigido por modelos (MDSD)?

¿Cómo responden los modelos para satisfacer los requisitos que exige la complejidad de las aplicaciones de software?

2.2 Criterios de inclusión y exclusión

Inicialmente se realizaron búsquedas de artículos relacionados con el MDD, que generaba una gran lista de resultados, pero que se fue filtrando hasta dar con el enfoque requerido que algunos trabajos tenían. Para la elaboración de esta revisión de literatura se tuvieron en cuenta las publicaciones creadas en los periodos de 2008 – 2013, que tratan sobre la problemática que estamos estudiando.

Los criterios de inclusión fueron determinados por la necesidad de analizar los riesgos en la calidad del software, que fueron citados en el resumen para su estudio y que puede ser soportado con el enfoque del desarrollo de software dirigido por modelos (MDSD).

Estas son las palabras claves que se utilizaron como cadenas de búsqueda para la revisión de literatura y extracción de la información sobre los riesgos que existen en el desarrollo de software que no tienen en cuenta metodologías de última generación:

- ✓ Models risk control AND (model-driven software development OR MDSD) AND risks in software development.
- ✓ (State of the art OR article (risk control in software development))
- ✓ "model driven development quality”

2.3 Trabajos seleccionados

Tabla 1. Trabajos seleccionados

ID	Título	Autores	Fuente
1	The Benefits of Model-Driven Development in Institutional Repositories	(Texier, José; De Giusti, Marisa; Oviedo, Néstor; Villarreal, Gonzalo L.; Lira, Ariel. 2010)	http://alarcos.esi.uclm.es/per/fruiz/cur/santander/avallecillo-dsdm.pdf

2	Using a functional size measurement procedure to evaluate the quality of models in MDD environments	(Beatriz Marín, Giovanni Giachetti, Oscar Pastor, Tanja E. J. Vos, and Alain Abran. 2013)	ACM
3	MDA y el papel de los modelos en el proceso de desarrollo de software	(Juan Bernardo Quintero, Raquel Anaya, 2008)	http://revista.eia.edu.co/articulos8/Art.10.pdf
4	Towards an architecture for ensuring product quality in model-driven software development	(Javier Gonzalez-Huerta, David Blanes, Emilio Insfran, and Silvia Abrahão. 2010)	ACM

3. Evaluación de los trabajos seleccionados

Para la elaboración de la evaluación de los trabajos seleccionados se tuvieron en cuenta varios aspectos que agrupan los criterios necesarios, para que un trabajo fuera escogido por su gran aporte, que nos aproxima a dar una solución a la problemática.

3.1 Criterios de evaluación

Los trabajos fueron evaluados teniendo en cuenta las siguientes características que ayudan en la extracción de la información, para la revisión:

Criterio 1: Implementación del análisis de riesgos en las diferentes etapas del desarrollo de software.

Criterio 2: Beneficios de Model-Driven Development.

Criterio 3: Modelos establecidos para gestionar, administrar e implementar la calidad del software.

Criterio 4: Atributos de calidad del software frente a enfoque MDD.

Criterio 5: Ventajas del desarrollo guiado por modelos de software.

3.2 Resultado de la evaluación

Tabla 2. Resumen de la evaluación

Criterios de evaluación						
		Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5
Trabajos	**ID1**	1	3	1	3	3
	ID2	2	4	5	5	4
	ID3	3	3	3	3	4
	ID4	2	4	5	5	4
Valor ponderado de 1 a 5, según aplicabilidad en el criterio.						

ID1: Vemos que se mencionan algunas características que benefician el desarrollo de software, como la generación automática de aplicaciones a partir de modelos. Los modelos sirven para razonar y validar el sistema, detectar errores y omisiones en el diseño. La utilización de modelos MDSD, permite la integración con sistemas existentes y la especificación de los requisitos del sistema independientemente de las plataformas de implementación. Protege la inversión ante los continuos cambios en las tecnologías y permite abordar mejor sistemas más complejos.

ID2: Este articulo usa procedimientos de medición para evaluar la calidad de los modelos en entornos MDD. Toma los modelos como artefactos clave, para garantizar la calidad del software, utilizando técnicas de lectura para la detección de defectos, donde son comparados y relacionados con la exactitud y la consistencia de los modelos.

ID3: El objetivo de este trabajo es lograr que el reúso se integre de forma sistémica en las diferentes etapas del desarrollo, de tal manera que su impacto en

los diferentes artefactos resultantes del proceso de desarrollo sea efectivo y, en lo posible, medible. Se describe una propuesta para el desarrollo dirigido por modelos, con base en la aproximación por MDD. El reúso de software es una de las estrategias que se considera promisoria para que la industria de software pueda enfrentar el reto de desarrollar productos con niveles de calidad y productividad adecuados en un contexto de negocio altamente complejo y dinámico y con acelerados cambios tecnológicos. Se describen los riesgos técnicos que son mitigados con la aproximación del desarrollo por modelos. Se describe el reúso del software como principal ventaja en el desarrollo por modelos.

ID4: Este trabajo presenta una arquitectura para realizar transformaciones de modelos impulsados por que estén orientadas por los atributos de calidad. El objetivo principal de la arquitectura es definir un conjunto de artefactos y un proceso para especificar y ejecutar transformaciones modelo en el que la selección de las transformaciones alternativas se realiza sobre la base de atributos de calidad. En concreto, nos centramos en cómo asociar los atributos de calidad de las diferentes transformaciones alternativas y cómo esta información puede ser tomada en cuenta en un proceso de transformación automatizada para obtener los artefactos de software con los atributos de calidad deseados.

4. Discusión

El reto que en la actualidad motiva a la comunidad de investigadores y generadores de tecnología es proponer esquemas de desarrollo en los cuales los modelos, antes que el código, son los actores centrales del proceso de desarrollo y donde se proveen mecanismos y herramientas de trabajo integradas que asisten al desarrollador en la construcción y transformación progresivas de modelos hasta llegar a la solución final. Por eso el objetivo del artículo **ID1** es explorar los principales conceptos que rigen MDD y presentar una propuesta para el proceso de desarrollo de software dirigido por modelos. Estos modelos dirigen el desarrollo software de un nivel de abstracción alto a otro nivel inferior, hasta llegar al código fuente. Permite realizar validaciones y verificaciones sobre los modelos e

identificar errores en fases tempranas del desarrollo como en la fase de diseño. Además, de poder anotar las características críticas (atributos de calidad) en los modelos. Según el artículo **ID2**, para producir software de alta calidad mediante el uso de métodos de MDD, la garantía de calidad de los modelos es de suma importancia, porque contribuye con la detección de defectos Actualmente, es ampliamente aceptado que el tamaño funcional de las aplicaciones es esencial para aplicar modelos de estimación, modelos de esfuerzo y modelos de presupuesto de proyectos que permiten al jefe de proyecto generar indicadores para facilitar la gestión de los proyectos. Por esta razón, este artículo se ha centrado en la medición del tamaño funcional, proponiendo un procedimiento de medición para medir el tamaño funcional de las aplicaciones generadas en entornos MDA.

En el artículo **ID3**, los diversos estándares en los que se apoya la propuesta de MDA tienen el propósito de lograr la interoperabilidad de las herramientas y plataformas, posibilitando evadir los problemas por la diversidad de plataformas y la evolución tecnológica que impregnan el código fuente que representa a los modelos software, presenta también serias consecuencias. En primer lugar, hace que la brecha existente entre modelos y código se incremente. Si en los modelos es sencillo separar el modelo funcional del modelo tecnológico de la aplicación, en el código fuente esta tarea se torna imposible. Esto hace que la única representación de los programas que puede ejecutarse, su código fuente, quede inexorablemente ligada a plataformas tecnológicas concretas. En un ámbito donde los artefactos tecnológicos se renuevan en cuestión de pocos años, este problema afecta directamente al mantenimiento y reutilización del software a medio-largo plazo. Por otra parte, este problema afecta directamente a la eficiencia y fiabilidad de la actividad codificación. Un código fuente donde se entremezclan módulos de funcionalidad correspondientes a diferentes dominios resulta difícil de manipular manualmente de una manera eficiente. Este paso a herramientas capaces de generar código, permitiendo concentrar la atención en la articulación de un modelo del problema a resolver, liberándose de las tareas repetitivas, de realizar malas estimaciones convirtiéndose en un riesgo especifico de la revisión y representa

una mejora sustancial que sin duda se extenderá, de una u otra forma a controlar toda clase de riesgos que se puedan presentar en el desarrollo de aplicaciones de software. Los generadores de código han desarrollado una historia, y no todos ellos quizá tengan el mismo valor o alcance, pero contribuyen a la consolidación de un concepto acerca de cómo crear y mantener software. Justamente la iniciativa MDA del Object Management Group (OMG), las ideas delineadas por Microsoft en torno a Software Factory, e incluso la creciente aparición de Lenguajes de Dominio soportando el concepto de Modelado Específico de Dominio, representan caminos de crecimiento en la construcción de software.

Para el artículo **ID4**, Se define una arquitectura para la transformación de modelos de calidad impulsada. El objetivo principal de la arquitectura es definir un conjunto de artefactos (compuesto principalmente por el modelo y metamodelos) y un proceso que permite la definición y ejecución de las transformaciones de modelos en los que la selección de las transformaciones alternativas son hechas sobre la base de atributos de calidad. El fundamento de este enfoque es ser capaz de seleccionar automáticamente la transformación alternativa que un desarrollador de software con experiencia aplicaría de forma manual.

5. Conclusiones

Una de las ideas que consideramos más prometedoras de este enfoque es lograr una separación del modelado del espacio del problema y los modelos del espacio de la solución, de tal manera que los detalles tecnológicos e ingenieriles que son irrelevantes para la funcionalidad del negocio se encuentren plasmados en un modelo de descripción de la plataforma que se puede fundir con el PIM para generar un PSM. Esta separación clara de los niveles de abstracción les permite a los desarrolladores enfocarse en el dominio del negocio y el cumplimiento de los requisitos más que en el dominio de la tecnología. El desarrollo de esta revisión ha resaltado una serie de aspectos importantes con relación al desarrollo de software dirigido por modelos, como los atributos de calidad del desarrollo de las aplicaciones. Debido a la creciente complejidad de estas aplicaciones, el campo

de la Ingeniería se ha desarrollado de manera muy acelerada. Pero desafortunadamente, dicha complejidad no está acompañada de los mecanismos adecuados que garanticen la calidad de las aplicaciones, cosa que si se puede lograr con el enfoque MDD.

La Evaluación de riesgos debiera centrarse en entregar función de negocio, no en el proceso de construcción, ya que de diversas formas, se puede perder mucho tiempo en lo menos importante, restándole utilidad al aplicativo. Para esto creo que se debe ajustar el proceso en lo que tiene que ver con la negociación del alcance con el cliente y el apoyo en las nuevas tecnologías que tienen una curva de aprendizaje elevada.

6. Referencias

Beatriz Marín, Giovanni Giachetti, Oscar Pastor, Tanja E. J. Vos, and Alain Abran. (2013). Using a functional size measurement procedure to evaluate the quality of models in MDD environments.*ACM Trans. Softw. Eng. Methodol.* 22, 3, Article 26 (July 2013), 31 pages. DOI=10.1145/2491509.2491520

Javier GONZALEZ-Huerta, David Blanes, Emilio Insfran, and Silvia Abrahão. (2010).Towards an architecture for ensuring product quality in model-driven software development. In*Proceedings of the 11th International Conference on Product Focused Software* (PROFES '10). ACM, New York, NY, USA, 28-31. DOI=10.1145/1961258.1961265

Trudel, S. and Abran, A. (2010). Functional requirements improvements through size measurement: A case study with inexperienced measurers. In Proceedings of the 8th ACIS International Conference on Software Engineering Research, Management and Applications (SERA'10). IEEE Computer Society, 181--189.

Juan Bernardo Quintero, Raquel Anaya. MDA y el papel de los modelos en el proceso de desarrollo de software, (2008). http://revista.eia.edu.co/articulos8/Art.10.pdf

Texier, José; De Giusti, Marisa; Oviedo, Néstor; Villarreal, Gonzalo L.; Lira, Ariel. (2010). The Benefits of Model-Driven Development in Institutional Repositories. http://alarcos.esi.uclm.es/per/fruiz/cur/santander/avallecillo-dsdm.pdf

FOWLER, Martin. Domain specific language. [Documento electrónico] MartinFowler.com, (2006). (Citada: 22 agosto 2006) http://www.martinfowler.com/bliki/Do- mainSpecificLanguage.html