
Ingeniería Dirigida por Modelos como soporte a la gestión de procesos software

Iván Ruiz-Rube, Juan Manuel Dodero y Mercedes Ruiz

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Cádiz, c/Chile nº1, Cádiz (España)
{ivan.ruiz, juanma.dodero, mercedes.ruiz}@uca.es

Resumen

Las iniciativas para la mejora de la calidad en el proceso software, suelen ser difíciles de implantar, por lo que es importante investigar nuevos métodos que permitan reducir esta dificultad. En este trabajo, se resume el estado del arte del soporte que ofrece la ingeniería dirigida por modelos (MDE) a la gestión de la calidad en los procesos software, desde la óptica de la gestión de procesos de negocio (BPM). Los resultados de la investigación han permitido comprobar que el uso de este enfoque es eficaz para la definición, ejecución y monitorización de los procesos software organizacionales.

1. Introducción

La gestión de la calidad (QM) tiene por objetivo planificar, asegurar, controlar y mejorar la calidad del software que se desarrolla o se mantiene.

El papel de la calidad en la Ingeniería del Software es vital en el éxito o el fracaso de los proyectos. Sin embargo, es común que no se le dediquen los esfuerzos y recursos necesarios, debido a que las actividades de QM se consideran como "no productivas", en el sentido que no conducen a generar nuevo software. Por este motivo, es importante investigar nuevas maneras de llevar a cabo exhaustivas actividades de calidad, con la menor interacción humana posible y por tanto, con el menor coste asociado.

Por otra parte, la Ingeniería Dirigida por Modelos (MDE) es el paradigma que abarca el conjunto de métodos, técnicas y tecnologías destinadas a construir software de forma más rápida y sencilla, mediante el desarrollo y transformación de modelos.

Podemos plantearnos la siguiente pregunta: ¿Sería posible utilizar el enfoque MDE para potenciar la calidad del software? La hipótesis de la investigación se basa en la idea de que *sí es posible usar MDE como mecanismo de soporte a las actividades de gestión de la calidad*.

Este trabajo resume el estudio realizado en el trabajo fin de máster [5], focalizándonos en el análisis de MDE como soporte a los procesos software, desde la óptica de la gestión de procesos de negocio (BPM).

El resto del trabajo se estructura de la siguiente manera: En el apartado 2 se resumen los

antecedentes necesarios para una correcta comprensión del problema de investigación. En la sección 3 se exponen los principales lenguajes y herramientas para el modelado de procesos software. Los resultados de la revisión de la literatura referente a la utilización de MDE en el contexto del ciclo de vida de los procesos de negocio, son recogidos en el punto 4. Finalmente, se presentan las conclusiones y las líneas de trabajo futuras.

2. Antecedentes

Los fundamentos bajo los que se sustenta el presente trabajo de investigación son la gestión de la calidad en el software y la ingeniería dirigida por modelos.

2.1. Gestión de la calidad

La gestión de la calidad puede contemplarse a dos niveles: proceso y producto. Es una premisa, que la calidad del software desarrollado (producto) depende de la calidad procesos ejecutados [2]. Por este motivo, los esfuerzos relativos a la calidad deben focalizarse en ambos niveles.

Dentro de la Ingeniería del Software, existe una disciplina conocida como la Ingeniería de Procesos Software, cuya motivación es la de crear un marco de trabajo técnico y de gestión, que permita la producción sistemática de software mediante procesos bien definidos.

Por otra parte, BPM es un campo que está levantando gran interés tanto en ámbitos empresariales como técnicos. Weske [7] definió a los procesos de negocio como aquel conjunto de actividades que se desarrollan en

coordinación y bajo un entorno organizacional y tecnológico, que permiten alcanzar un determinado objetivo de negocio.

La idea subyacente en BPM es permitir a las organizaciones definir sus actividades en términos de sus procesos de negocio, ya sean de desarrollo o mantenimiento de productos, o prestación de servicios.

En este punto, podríamos considerar los procesos software como un tipo particular de procesos de negocio. Así, los procesos software siguen también el ciclo de vida de los procesos de negocio, representado en la figura 1.

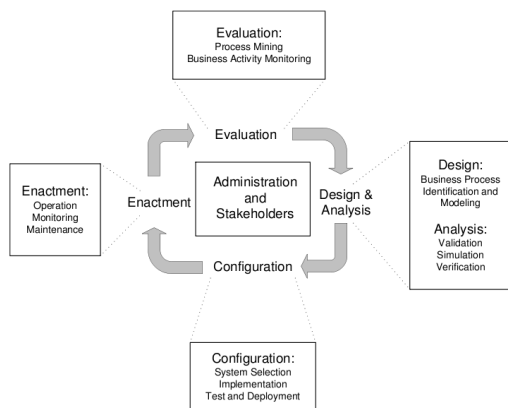


Figura 1. CV de un proceso de negocio (Weske).

2.2. Ingeniería Dirigida por Modelos (MDE)

MDE es la disciplina dentro de la Ingeniería del Software que tiene por objetivo dar soporte a las actividades del ciclo de vida del software, utilizando modelos como principal artefacto de desarrollo.

En este enfoque destaca la propuesta de la Object Management Group (OMG) para el desarrollo de modelos, MDA [3], basada en una arquitectura de modelado, una serie de niveles de abstracción y un conjunto de estándares y tecnologías, como se representa en la figura 2.



Figura 2. Visión de la OMG para MDE.

3. Lenguajes y Herramientas

Definir los procesos software utilizando modelos procesables por el ordenador, nos

ofrece la posibilidad de aprovechar las oportunidades que ofrece la ingeniería dirigida por modelos, con el fin de dar soporte a la ejecución y evaluación de los procesos implantados [4].

La definición de las descripciones de procesos se enmarca dentro del área de proceso *Organizational Process Definition (OPD)* de CMMI [6] y en *Process Improvement Process Group (PIM)* de ISO/IEC 12207 [1].

3.1. Lenguajes de modelado de procesos (PML)

Lenguajes gráficos como *BPMN* o los diagramas de actividad de *UML*, permiten construir descripciones de procesos, de forma homogénea.

Estos lenguajes ayudan a mejorar la correcta comprensión de los procesos por parte de los implicados en los mismos. Sin embargo, con el objetivo de modelar y representar más fielmente los procesos software, se han desarrollado varios lenguajes específicos, que comparten ciertas características comunes, como la representación de actividades, recursos, productos de trabajo, actores responsables y reglas implícitas al proceso.

A continuación, describimos los lenguajes más significativos:

- *SPEM* es el lenguaje propuesto por la OMG, para la definición de procesos software.
- *SEMDM* es la alternativa para el modelado de procesos de desarrollo software, definida en el estándar ISO/IEC 24744.
- *MSF* es el marco de trabajo que propone Microsoft para definir procesos software.
- *OPF* es una propuesta libre, de dominio público y basada en la industria, para la producción sistemática de metodologías.

3.2. Herramientas de soporte

Para una correcta definición de los modelos de procesos, es preciso disponer de herramientas de soporte a los lenguajes de modelado. Podemos destacar las siguientes:

- *EPF Composer* es la herramienta de la comunidad Eclipse, para la creación, edición y mantenimiento de fragmentos de métodos, procesos o metodologías, basadas en el metamodelo SPEM.
- *Objectteering*, de Objectteering Software, es una herramienta de soporte a MDE, que permite además, modelar procesos software utilizando el estándar SPEM.
- *Enterprise Architect*, de Sparx Systems, permite el análisis y diseño de distintos tipos de modelos, incluyendo aquellos basados en SPEM.

- *IRIS Process Author*, de Osellus Inc., es una solución para la autoría y monitorización de procesos software compatibles con SPEM.
- *Visual Studio ALM*, de Microsoft Corporation, es una solución de soporte al ciclo de vida del software. Dentro del paquete *Power Tools* existe una aplicación para editar plantillas de procesos MSF.

4. Ciclo de vida

En la literatura científica, se localizaron un total de 64 publicaciones posteriores a 2004 referentes al tema de investigación, utilizando para ello, los buscadores *Google Scholar*, *TDG Scholar* e *Inspec*. Finalmente, fueron 51 las publicaciones analizadas y clasificadas, entre propuestas, metodologías y experiencias relativas a la utilización de modelos durante las actividades enmarcadas dentro del ciclo de vida de los procesos software.

En la figura 1 podemos comprobar el volumen de contribuciones científicas de interés, distribuidas según la fase del ciclo de vida a la que dan soporte y el lenguaje de modelado utilizado.

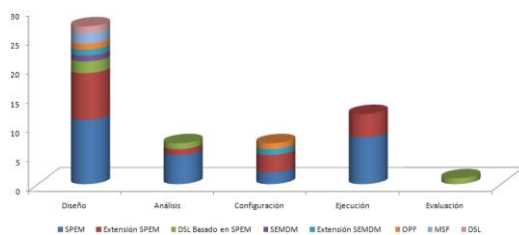


Figura 3. Contribuciones según el ciclo de vida.

Tras el estudio de estos trabajos, podemos extraer las siguientes conclusiones:

4.1. Diseño de procesos

El ciclo de vida de los procesos software comienza con el diseño de los procesos, haciéndolos explícitos mediante la utilización de los lenguajes destinados a tal efecto (PMLs). En la literatura hemos observado distintas aplicaciones de los PMLs para el modelado de metodologías ágiles (OpenUP, XP o Scrum), metodologías regionales como *Metrica 3* o *V-Modell*, metodologías orientadas a agentes, marcos de mejora (como *CMMI* o *ISO 12207*) y otros enfoques.

El lenguaje más utilizado por los autores para el diseño de los modelos de proceso es SPEM. Sin embargo, se han planteado numerosas extensiones y lenguajes derivados del mismo, con el objetivo de cubrir carencias concretas que presenta el lenguaje, fundamentalmente en lo referente a la ejecución de los procesos.

4.2. Análisis de procesos

Los procesos software descritos mediante algún PML deben evaluarse para comprobar las bondades y deficiencias que presentan. Al utilizar modelos como base para la definición de los procesos software, es posible reutilizar las técnicas habituales para el análisis de modelos. En la investigación, hemos encontrado estudios relativos a:

- *Verificación*: transformación en Redes de Petri y definición de reglas en OCL.
- *Validación*: usabilidad de modelos, análisis visual desde distintas perspectivas y la adecuación con respecto a marcos de referencia, como *CMMI*.
- *Simulación*: transformación de modelos conceptuales en modelos interpretables en entornos de simulación, como *DEVS-Hybrid* o *SimSe*.
- *Métricas*: obtención de medidas e indicadores de calidad desde modelos de proceso.

4.3. Configuración de procesos

Esta fase contempla la selección del sistema de gestión de procesos y la configuración del mismo según las características de la organización y de los procesos que se desplegarán en él. Es posible utilizar los sistemas de gestión de procesos de propósito general existentes, habitualmente diseñados según una arquitectura basada en servicios (SOA).

En el estudio realizado, hemos podido descubrir algunas propuestas académicas destinadas a la creación entornos software (habitualmente basados en Eclipse) que permitan diseñar procesos, desarrollar modelos y gestionar las instancias de los procesos.

Por otra parte, es difícil encontrar organizaciones que utilicen una misma metodología para dar soporte a la ejecución de sus proyectos. Esto se debe a la naturaleza variable del proceso software: “no existen dos proyectos iguales”.

Así, está tomando gran importancia la idea de modelar la variabilidad implícita en los procesos (líneas de procesos software) y la Ingeniería de Métodos Situacional. Esto es, la creación de fragmentos de método, en lugar de metodologías completas, que puedan ser luego adaptadas (*tailoring*) a las necesidades concretas de los proyectos.

4.4. Ejecución de procesos

Una vez finalizada la fase de configuración, se deben instanciar (*enactment*) los procesos software con el comienzo de cada proyecto. El uso de lenguajes y sistemas basados en

workflows son los mecanismos más comunes para la secuenciación de tareas y la asignación de responsabilidades a roles concretos.

Existen trabajos relativos a transformaciones desde modelos SPEM (o de sus derivados) a modelos de workflows como BPEL o XPD. Estas transformaciones tienen por objetivo la orquestación automática de servicios en plataformas de integración de procesos. También se recogen trabajos relativos a transformaciones desde modelos de procesos a documentos web y plantillas MS Project.

Con respecto a la monitorización de procesos software, se ha estudiado la utilización de reglas definidas sobre ontologías de la web semántica, las cuales permiten encontrar inconsistencias entre la definición de los procesos y los procesos finalmente ejecutados.

4.5. Evaluación de procesos

Para la evaluación de los procesos de negocio, se utilizan herramientas para la monitorización en tiempo real y/o análisis post-mortem, las cuales suelen estar integradas en los sistemas BPM. Desde el punto de vista de los procesos software, existe una propuesta denominada SPAGO4Q que permite evaluar procesos, mediante la obtención y análisis de métricas recopiladas de distintas herramientas de soporte al desarrollo.

5. Conclusiones

La mejora de la calidad en el software es uno de los aspectos más importantes y demandados por los usuarios finales y en consecuencia por todos los implicados en el éxito de los proyectos de desarrollo o mantenimiento software.

Por otra parte, el uso del paradigma MDE promueve el desarrollo de modelos, como elemento fundamental en las distintas disciplinas de la Ingeniería del Software.

Este trabajo recopila las principales conclusiones del trabajo fin de máster [5] orientado al estudio de MDE como soporte a la gestión de la calidad en el software.

Sin embargo, en lugar de estudiar el soporte de MDE respecto de algún marco específico para la mejora de procesos (CMMI o SPICE) para evitar obtener resultados parciales, hemos optado por estudiar el soporte desde la óptica de la gestión de los procesos de negocio.

En la revisión de la literatura, hemos descubierto un buen número de trabajos que basándose en lenguajes de procesos software (PMLs), como SPEM, modelan metodologías o marcos de referencia para la mejora de procesos. Además, se han localizado distintas propuestas enfocadas a la validación,

configuración, ejecución y evaluación de procesos software.

Así pues, podemos afirmar que MDE es un enfoque adecuado para dar soporte a la mejora de los procesos software, desde la perspectiva de la gestión de los procesos de negocio.

6. Trabajo Futuro

A continuación se describen algunas de las posibles líneas de investigación abiertas.

Modelado de aspectos no funcionales en los procesos software, que no estén estrechamente relacionados con la secuenciación y ejecución de tareas. Utilizando un enfoque basado en la *separación de conceptos*, sería posible modelar aspectos de aprendizaje humano y de aseguramiento de la calidad en los procesos.

Estudiar las posibilidades que ofrece la propuesta de la OMG para la *modernización* de aplicaciones ADM, desde el punto de vista de los procesos software.

Exploración de nuevos mecanismos para la *simulación* de procesos desde modelos de procesos definidos con SPEM.

Profundización en el modelado de *líneas de procesos* software.

Agradecimientos

Esta investigación ha sido financiada por el Ministerio de Industria, Turismo y Comercio, mediante el proyecto eCultura (TSI-020501-2008-53) del programa Avanza I+D.

Referencias

- [1] I.S.O.: *Systems and software engineering. Software life cycle processes* (2008)
- [2] Juran, J.: *Quality in Software Development*. McGraw-Hill Professional (1998)
- [3] Kleppe, A. G., Warmer, J., Bast, W.: *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison-Wesley Longman Publishing Co., Inc.(2003)
- [4] Ruiz, F.: *Software Process Engineering: De una gestión de procesos Contemplativa a una Productiva*. Alarcos UCLM (2007)
- [5] Ruiz-Rube, I, Escalona M.J.: *Gestión de la calidad usando Ingeniería Dirigida por Modelos*. Trabajo Fin de Máster. Universidad de Sevilla. (2010)
- [6] S.E.I.: *CMMI for Development v. 1.2* (2006)
- [7] Weske, M.: *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag New York Inc (2007)

ⁱ Por motivos de espacio, no se incluyen las referencias a los trabajos localizados en la revisión de la literatura