

Implementación de un modelo de validación para la definición y mantenimiento de procesos de desarrollo de software

Claudio Javier Gonzalez, Maria Paula Izaurralde, Gonzalo Garcia Favre, Diego Cohen, Alvaro Ruiz de Mendarozqueta

{cjgonzalez, mizaurralde, 35369, 48955, aruiz}@sistemas.frc.utn.edu.ar
Laboratorio de Investigación en Ingeniería y Calidad de Software
<http://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/>
Departamento de Ing. en Sistemas de Información
Universidad Tecnológica Nacional
Maestro M. López esq. Cruz Roja Argentina
(X50165ZAA) Ciudad Universitaria, Córdoba, Argentina

Abstract. Las empresas de desarrollo de software, en búsqueda de mejorar su rendimiento y la calidad de sus productos, implementan modelos y normas como CMMI e ISO. Para sostener la mejora en el tiempo, es necesario modificar los procesos continuamente para adaptarlos al contexto cambiante. Los cambios en el proceso pueden incurrir en el incumplimiento de una norma o modelo en los que el proceso fue basado. Por ello es importante crear, mantener y evaluar mapeos entre el proceso y los modelos. Estos mapeos son útiles al momento de la validación de los procesos respecto de los modelos. Una dificultad que se presenta al momento de la validación es que los requerimientos de los modelos están descritos en lenguaje natural quedando librados a la interpretación de un experto. Este trabajo presenta un metamodelo que permite definir en forma más precisa los requerimientos de los modelos y las normas y un mecanismo de mapeo y validación de procesos.

Keywords: Validación - Modelo - Proceso – Metamodelo.

1. Introducción

En búsqueda de optimizar su rendimiento, las empresas de desarrollo de software definen sus procesos de desarrollo estandarizando sus actividades para luego mejorarlas [1] [2]. Una fuente para lograr esas mejoras consiste alinear sus procesos de desarrollo a normas y modelos, como CMMI [2] e ISO [3], cuyo objetivo es la mejora continua de los procesos de desarrollo de software [16].

A lo largo del tiempo, los procesos definidos por las empresas son modificados con distintos fines. Algunas modificaciones buscan alcanzar una mejora en el rendimiento del proceso y otras son debidas a la adaptación al contexto cambiante donde se desenvuelven las empresas en la actualidad. Las normas y modelos no están exentas a estos cambios, las mismas evolucionan agregando, modificando o quitando algunos

de sus elementos. Cualquiera sea el caso, éstas mejoras pueden incurrir en el incumplimiento de un modelo o norma en la que el proceso estaba basado.

Por lo tanto, es prudente evaluar el impacto de los cambios de los procesos y las normas para evitar debilitar el alineamiento entre ambos. Con ese fin, se recomienda la creación de documentos que indiquen la relación entre los elementos del proceso y los requerimientos de las normas o modelos. En CMMI estos documentos son llamados PIIDs (Process Implementation Indicator Descriptions).

Es de gran importancia que los documentos que describen las relaciones entre los modelos y los procesos organizacionales sean actualizados a medida que se realizan modificaciones al proceso organizacional, de este modo, se valida el impacto del cambio y se logran mantener las relaciones previamente establecidas [4].

Los procesos de creación, actualización de los mapeos del proceso respecto de las normas o modelos y la evaluación del impacto de los cambios es realizado principalmente por especialistas de procesos. Esta tarea es escasamente soportada por herramientas por lo cual es principalmente ejecutada en forma manual, tornándose en una tarea costosa y quedando expuesta a errores [19].

Otro punto que agrega complejidad a esta tarea, es que los modelos y normas, junto a los requisitos que se deben cumplir para lograrlos, están expresados en lenguaje natural y en diferentes formatos, dependiendo de la interpretación de un experto. Podemos inferir que para poder evaluar el cumplimiento de algún requisito de un modelo o norma, se deben definir de manera no ambigua las condiciones que se deben cumplir para un elemento que se desee validar.

Estos factores impactan directamente en el costo, efectividad y el riesgo de los proyectos de mejora de procesos organizacionales que las empresas deben implementar para lograr las mejoras deseadas.

Por otra parte, los proyectos de desarrollo de software, utilizan los procesos y los adaptan a sus necesidades específicas, en otras palabras, lo adaptan al problema que se planea resolver. Estos cambios, en consecuencia, potencialmente podrían desalinear la ejecución del proceso de la definición, y por lo tanto, de los modelos y normas en los que los procesos fueron basados [6]. Vemos así, que es importante también definir un mecanismo que permita evaluar el impacto de cambios de manera continua tanto en la definición como en la ejecución de los procesos organizacionales.

Estas problemáticas fueron el objetivo de investigación del proyecto de investigación "Implementación de un modelo de validación de procesos de desarrollo de software" [15] que ha dado origen a este trabajo.

1.1 Estado del arte

Como se ha mencionado anteriormente, la ambigüedad en la definición de los elementos requeridos en modelos de mejora de procesos de desarrollo agrega dificultad al mantenimiento y validación de los mismos. Estas dificultades, son agravadas si los procesos de desarrollo están definidos en lenguaje natural, sin formato ni terminología estándar, sino que por el contrario, en terminología y formatos propietarios, que difieren entre las diferentes empresas.

Con el fin de reducir los problemas en la definición de procesos, se crearon meta-modelos, como SPEM [7], que proveen los conceptos necesarios para modelar, documentar, presentar, administrar, intercambiar y publicar métodos y procesos de

desarrollo. Además, en los últimos años aparecieron en el mercado herramientas que, basándose en los meta-modelos, permiten la definición de procesos de manera estándar [8] [9] [10] [11], impulsadas por empresas líderes en la industria del software.

En [12] se plantea el análisis comparativo y la selección de herramientas de modelado de proceso de desarrollo de software. Este análisis comprende la definición de un conjunto de requerimientos básicos que deben cumplir las herramientas, junto con un método heurístico de decisión para la selección de la herramienta que mejor se adapta a las necesidades de un proyecto.

En el área de validación de procesos, se propone la validación de procesos basados en CMMI a través de UML [13] [14]. Estos trabajos se enfocan a definir, mediante UML, los elementos del modelo CMMI y de los procesos que fueron basados en el modelo para luego validarlos a través del uso de OCL [5]. Para lograr la validación, se debe expresar primero el proceso de una empresa en la terminología de UML propuesta y definir las restricciones de OCL que definen los mapeos entre CMMI y el proceso. Si bien se hace foco en procesos basados en CMMI se plantean mecanismos de extensión para poder cubrir otros modelos o normas. Con este enfoque la validación de los procesos se realiza de forma automática pero el esfuerzo de expresar el proceso en UML es significativo. Además, los analistas de proceso debes estar familiarizados con la metodología propuesta.

1.2 Propuesta

El proyecto de investigación “Implementación de un modelo de validación de procesos de desarrollo de software”, [15] basándose en la problemática descrita con anterioridad, propuso la creación de un metamodelo de validación de procesos cuyos conceptos permitan definir de manera no ambigua los elementos y condiciones requeridas por una norma o modelo, las relaciones de mapeo existentes con el proceso organizacional y proveer definiciones precisas de cumplimiento de un modelo o norma de mejora de procesos.

La propuesta también busca sentar las bases para la implementación de una herramienta de software que soporte las actividades de evaluación de impacto de los cambios y creación de mapeos de las normas y modelos con los procesos organizacionales.

El metamodelo de definición de elementos requeridos de una norma contiene los conceptos necesarios para identificar los elementos requeridos respecto de los cuales se valida el cumplimiento. El propósito de este metamodelo, es eliminar la ambigüedad en la definición de los elementos requeridos de una norma o modelo de mejora. En adelante se describirán los conceptos que componen el metamodelo en detalle.

2. Metamodelo de definición de elementos requeridos

Objetivo de validación (o):

Representa un objetivo o requerimiento que se logra a través de la implementación de un conjunto de elementos del modelo o norma en el proceso organizacional. Es un concepto generalmente abstracto que abarca o agrupa un conjunto de prácticas de una

disciplina de la ingeniería de software. Este concepto puede ser usado para representar un nivel de madurez o capacidad de CMMI.

Requerimiento de validación (v):

Es un conjunto de elementos del modelo que comparten una condición de cumplimiento para el objetivo de validación al que pertenecen. Los elementos del conjunto deben ser todos requeridos, al menos uno requerido u opcionales para que se cumpla la categoría de validación. En CMMI un conjunto de objetivos específicos de un nivel de capacidad de un área de procesos puede ser representado por un conjunto de elementos de validación, en ese caso, serían un conjunto de elementos requeridos para el cumplimiento de un nivel de madurez de un área de proceso. El conjunto de prácticas genéricas y específicas de un nivel de madurez de un área de procesos, podría representarse también con un conjunto de elementos de validación, pero en este caso el conjunto agruparía elementos opcionales.

Condición de cumplimiento (ζ):

Es un conjunto finito y conocido de elementos usados como calificadores para los grupos de elementos de validación. Los elementos del conjunto son “al menos una requerida” (ζ_u), “todas requeridas” (ζ_r), “opcionales” (ζ_o). El elemento “al menos una requerida” significa que al menos uno de los elementos del grupo de elementos de validación debe cumplirse por el proceso para que se cumpla el conjunto de elementos de validación, representa elementos suplementarios. Por otra parte, el elemento “todas requeridas” significa que todos los elementos del grupo de elementos de validación deben ser implementados por un proceso para cumplir el elemento de validación. Finalmente, el elemento “opcionales” significa que los elementos del grupo de validación tienen un fin opcional o informativo en la norma o modelo y no es necesario que el proceso las materialice para que se cumpla el grupo de elementos de validación.

Elemento de validación (ϵ):

Representa un elemento del modelo que se deba cumplir.

2.1 Operaciones entre elementos del modelo

2.1.1 Sobre escrituras de reglas de validación

$$S(s_i, j) = s_j / s_i, s_j \in v, j, i \in \zeta$$

La operación $S(s_i, j)$ se aplica sobre un objetivo de validación s_i , obteniendo como resultado un objetivo de validación s_j la cual contiene los mismos conjuntos de elementos de validación que s_i , pero con las condiciones de cumplimiento i reemplazadas por las condiciones de cumplimiento j . Esta operación, es útil a los fines de expresar situaciones en que los conjuntos de elementos de validación, de un objetivo de validación, cambien su condición de cumplimiento en diferentes contextos. Podemos encontrar un ejemplo de estas situaciones en el modelo CMMI, cuando se evalúa si una organización ha alcanzado el nivel de madurez 3, es requerido que al menos una área de proceso tenga el nivel de capacidad 3 y las restantes hayan

alcanzado el nivel de capacidad 2. Luego si se evalúa el nivel de madurez 4, es requerido que al menos una de las áreas de proceso haya alcanzado el nivel de capacidad 4 y que las restantes hayan alcanzado el nivel de capacidad 3. En estos ejemplos, los objetivos de validación mantienen sus grupos de elementos de validación constantes, pero es necesario cambiar la condición de validación en las diferentes situaciones de validación. Los grupos de elementos de validación que en el nivel de madurez 3 tenían la condición de validación “al menos una requerida” deberán ser cambiados a “todos requeridos” para cumplir el nivel de madurez 4.

2.1.2 Unión de objetivos de validación

$$S_i \cup S_j = S_{ij} / S_i \subset S_{ij} \wedge S_j \subset S_{ij}$$

Dados los objetivos de validación S_i y S_j , la unión de objetivos de validación da como resultado un nuevo objetivo de validación que incluye los conjuntos de elementos de validación de S_i y S_j .

Esta operación es útil para representar situaciones como por ejemplo los niveles de madurez de CMMI en donde un nivel incluye los requerimientos de los anteriores.

2.1.3 Relación de mapeo ($R_m(p_1, \dots, p_n)$)

La relación de mapeo que representa la implementación de un elemento de la norma o modelo por parte de uno o más elementos del proceso. El dominio de la relación está compuesto por los elementos de un proceso y la imagen es un subconjunto de los elementos de la norma o modelo.

2.1.4 Enunciado de cumplimiento de una norma o proceso

$$\text{Proceso } p \text{ cumple Modelo } m \Leftrightarrow \forall \varepsilon_i \in v_{gr}, \forall v_{qu} \exists \varepsilon_j \in v_{qu} / \varepsilon_i, \varepsilon_j \in R_{R_m(p)}$$

Decimos que un proceso cumple una norma o modelo sí y solo sí, todo elemento del modelo incluido en un requerimiento de validación con condición de cumplimiento “todos requeridos” está incluido en el conjunto rango de la relación de mapeo y al menos uno de los elementos de cada requerimiento de validación con condición “al menos uno requerido” está incluido en la imagen de la relación.

3. Implementación del metamodelo de validación de procesos

Como parte del trabajo realizado por el laboratorio de investigación, se desarrolló una herramienta de software que implementa el metamodelo, permitiendo así la validación de procesos de desarrollo de software. Tomando como base la herramienta de definición de procesos Eclipse Process Framework Composer [11], se desarrolló una herramienta de software que permite instanciar los elementos del metamodelo de validación, definir mapeos y ejecutar las validaciones a un proceso. Esta herramienta se desarrolló con el fin de ser una prueba de conceptos del metamodelo. Se tomó como modelo de base CMMI, y se realizaron mapeos de éste con un proceso de desarrollo genérico.

3.1 Objetivos de la prueba de concepto

Para llevar a cabo la prueba de concepto, elegimos validar un proceso respecto al modelo CMMI, se optó por la representación continua, el área de proceso seleccionada fue la Gestión de la Configuración y se pretendió conocer el nivel de capacidad de un proceso dado.

El objetivo concreto de esta experiencia fue validar que un proceso dado satisface el nivel de capacidad 2 del área de proceso Gestión de la Configuración según lo establecido en el modelo CMMI.

A continuación presentamos un extracto del archivo de definición de los elementos a validar del modelo, correspondiente a los niveles de capacidad 1 (*Realizado*) y 2 (*Administrado*) respectivamente. Este archivo será referenciado en el ejemplo siguiente.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <validacion>
    <objetivoDeValidacion identificador="CM_C1"
      nombre="Gestión de la Configuración - Capacidad 1"
      modeloOrigen="CMMI" descripcion="Realizado">
      <requerimientoDeValidacion nombre="Metas de
        la Gestión de la Configuración" tipo="todas
        requeridas">
        <elementoDeValidacion
          identificador="CM.SG_1" />
        <elementoDeValidacion
          identificador="CM.SG_2" />
        .....
      </requerimientoDeValidacion>
      <incluyeObjetivoDeValidacion id="CM_C1">
        <sobreescribe de="al menos una requerida"
          a="todas requeridas" />
      </incluyeObjetivoDeValidacion>
    </objetivoDeValidacion>
  </validacion>
```

3.2 Objetivo de la validación (o)

Durante las evaluaciones del modelo, los evaluadores desean conocer los niveles de capacidad de las áreas de proceso. Entonces podemos definir a cada nivel de capacidad de un área de proceso como un objetivo de validación diferente. En el archivo XML presentado con anterioridad, identificamos a cada nivel de capacidad con las siglas del área de procesos seguida del nivel de capacidad. El elemento “objetivoDeValidacion” representa a un objetivo de validación en el archivo XML. Cada objetivo de validación cuenta además con un nombre, el modelo que valida y una descripción.

3.3 Requerimiento de validación (v) y Condición de cumplimiento (c)

El elemento “requerimientoDeValidacion” representa al componente del metamodelo requerimiento de validación en el archivo XML.

En el modelo CMMI, un nivel de capacidad consiste de metas y prácticas genéricas relacionadas a un área de proceso. Pero las áreas de proceso también están compuestas de metas y prácticas específicas, que corresponden a una única área de proceso y describen las características que deberían ser implementadas para satisfacer el propósito de la misma. Las metas genéricas y específicas son componentes requeridos en CMMI [18]. El cumplimiento de las mismas es analizado durante las evaluaciones para conocer el nivel de capacidad de las áreas de proceso. Las prácticas genéricas y específicas en cambio, son componentes esperados, ya que su sentido es guiar a los individuos y grupos tanto en la implementación de las mejoras como en la realización de las evaluaciones.

En la representación en XML, para diferenciar a los componentes requeridos de los esperados, hemos utilizado un atributo al que llamamos “tipo” que representa las condiciones de validación. Dado que las metas genéricas y específicas son elementos requeridos para poder alcanzar un nivel de capacidad, se ha definido un requerimiento de validación para cada nivel. Estos requerimientos de validación son del tipo “todos son requeridos” ya que para considerar a un área de proceso en un nivel de capacidad dado, es preciso que todos los elementos definidos en el requerimiento se hayan satisfecho. Es decir, que todos los elementos que pertenecen a este conjunto deben estar mapeados en el proceso para considerar que el objetivo de validación se ha cumplido.

Del mismo modo, se definieron requerimientos de validación para representar a las prácticas genéricas y específicas, y para diferenciarlas de los elementos requeridos, se las ha clasificado como “opcionales”.

Por otro lado, para asegurar que un área de proceso posee un nivel de capacidad n , es necesario primero que la misma cumpla con las metas del nivel anterior ($n-1$). Para representar esta característica del modelo y facilitar la definición de los validables, se incluyó un nivel más dentro de los requerimientos de validación, el cual permite a un objetivo de validación heredar otros requerimientos de validación y tomarlos como propio. Puede referirse a (Unión de objetivos de validación) para mayor detalle.

3.4 Elementos de validación (ϵ)

Las metas (genéricas y específicas) y prácticas (genéricas y específicas) han sido representadas por medio de los elementos de validación. Este componente está representado por el elemento “elementoDeValidacion” en el archivo XML.

3.5 Instanciación de relación de mapeos

La relación de mapeo se implementa a través archivos XML que relacionan elementos del proceso y del modelo. Es posible, que un elemento del modelo sea realizado completamente por un elemento del proceso, sea realizado completamente por varios elementos del proceso suplementariamente, o sea realizado por un conjunto de elementos de proceso en forma complementaria. Por ello los archivos XML fueron diseñados para poder expresar este tipo de relaciones.

A continuación, se presenta un ejemplo de los archivos XML usados para representar los mapeos.

```

<?xml version="1.0" encoding="UTF-8"?>
<processMapping processId="CMMI" mappedModelId="CMMI">
  <elementMapping elementId="CM.SG_1">
    <mappingGroup>
      <processElement id="CILOC_Labels"
        description="Esta sección
específica las etiquetas usados para un línea base,
establece el formato de las etiquetas a aplicar en cada
ocasión, para versiones borrador y para versiones línea
base."/>
      <processElement
id="CILOC_ConfigurationItems"
        description="Esta sección
identifica los ítems de configuración y el formato con el
que deben nombrarse."/>
    </mappingGroup>
  </elementMapping>
  .....
</processMapping>

```

3.6 Resultados de ejecución de herramienta

El resultado de la herramienta de validación de procesos es un reporte que determina si un proceso cumple el modelo objeto de evaluación. Además, incluye el detalle de los objetivos de validación que no se cumplen y la causa del incumplimiento. A continuación mostraremos ejemplos de los posibles resultados de la validación.

En el siguiente ejemplo podemos ver el resultado de una validación de proceso no exitosa en la que alguno de los elementos requeridos no se cumplen.

```

[Configuration Management Goals - AllRequired] Element
Mapping Missing: CM.SG_1
[Configuration Management Goals - AllRequired] Element
Mapping Found: CM.SG_2
[Configuration Management Goals - AllRequired] Element
Mapping Missing: CM.SG_3
[Configuration Management Goals - AllRequired] Element
Mapping Missing: CM.GG_1
The process Agile does not meet Configuration
Management - Capability 1. Some required elements are
missing
Overall status: ERROR

```

El resultado de la herramienta en el siguiente ejemplo muestra la salida para una validación exitosa. Podemos observar que todos los requerimientos de validación requeridos están implementados en el proceso evaluado.

```

[Configuration Management Goals - AllRequired] Element
Mapping Found: CM.SG_1
[Configuration Management Goals - AllRequired] Element
Mapping Found: CM.SG_2

```



```
[Configuration Management Goals - AllRequired] Element
Mapping Found: CM.SG_3
[Configuration Management Goals - AllRequired] Element
Mapping Found: CM.GG_1
The process CMMI meets Configuration Management -
Capability 1 but some optional elements are not met.
Overall status: SUCCESSFUL
```

4. Conclusiones

El metamodelo presentado en este trabajo provee una base teórica para reducir o eliminar los problemas relacionados con la validación de procesos respecto de modelos de mejora. Si bien el ejemplo usado fue el modelo CMMI, el metamodelo de validación fue diseñado para expresar requerimientos de validación de otros modelos y normas como ISO.

El desarrollo de la herramienta basada en el metamodelo a modo de prueba de conceptos, nos permitió validar los conceptos presentados y el enunciado de cumplimiento de una norma o modelo.

Los elementos del metamodelo permitieron a especialistas en CMMI describir los requerimientos de la norma en forma precisa. A su vez, las relaciones de mapeos permitieron expresar la instanciación de los elementos del modelo por parte de los elementos del proceso. Los elementos del metamodelo permitieron reducir o eliminar la ambigüedad en la definición de requerimientos y mapeos.

Luego, se ejercitó el enunciado de cumplimiento por medio de la implementación del mismo como una función de validación en un módulo de la herramienta de software. Se pudo observar que lograba determinar el cumplimiento, o no, del modelo para diferentes mapeos de procesos.

El metamodelo agrega información a la definición de una norma o modelo para lograr expresar con claridad los requerimientos de validación minimizando interpretaciones ambiguas.

5. Trabajos futuros

Actualmente el laboratorio de investigación se encuentra trabajando en la implementación de una herramienta de software que materialice el metamodelo y provea un ámbito de colaboración entre la universidad y empresas del medio. Expertos en las normas y modelos, pertenecientes a la universidad, crearán las definiciones de los modelos y normas en términos del metamodelo. Estas definiciones, estarán disponibles por medio de la herramienta para que las empresas puedan mapear sus procesos. En consecuencia, la herramienta dará a las empresas la capacidad de ejecutar validaciones. Las definiciones de requerimientos serán mantenidos por los expertos de la universidad, y en caso de que los modelos y normas cambien, las empresas no deberán invertir esfuerzo en esas actividades y podrán conocer el delta de sus procesos y los requerimientos de la norma o proceso. A su vez, la universidad podrá recabar información útil para determinar las prácticas de ingeniería con mayores necesidades según su nivel de mapeo o cumplimiento en las empresas locales.

Otra de las líneas de investigación, se inicia en el uso del metamodelo conjuntamente con herramientas de modelado e instanciación de procesos. Incluso, podría ser integrado con herramientas actuales que permiten la ejecución de procesos definidos, entonces la validación podría extenderse a los procesos ejecutados por los proyectos de desarrollo validando las adaptaciones que son realizadas para adaptar el proceso para lograr la resolución de un problema específico. De estas premisas surge la hipótesis de que es posible generar un modelo que permita la automatización total o parcial de la validación y control de cambios en las instanciaciones del proceso organizacional definido utilizando una herramienta de ALM (Administración de los ciclos de vida de las aplicaciones) base de futuros trabajos del proyecto de investigación y del laboratorio de investigación y desarrollo de ingeniería y calidad de software LIDICALSO.

6. Reconocimientos

El trabajo presentado en esta publicación fue desarrollado como parte del trabajo llevado a cabo por el laboratorio de investigación y desarrollo en ingeniería y calidad de software perteneciente a la “Universidad Tecnológica Nacional Córdoba”. Muchas personas además de los autores participaron en las investigaciones asociadas. En particular, nos gustaría agradecer a Diego Rubio, Natalia Andriano y Marcela Garay Moyano por sus grandes contribuciones al presente trabajo.

7. Referencias

1. **Humphrey, Watts S.** *Managing the Software Process*. Massachusetts : Addison-Wesley, 1989. ISBN: 0201180952.
2. **CMMI Product Team.** *CMMI for Development, version 1.2*. Pittsburgh, Pennsylvania, USA : Software Engineering Institute (SEI), August 2006. CMU/SEI-2006-TR-008.
3. **International Organization for Standardization.** ISO. [Online] ISO, 2008. <http://www.iso.org/iso/about.htm>.
4. **Pomietto, Robert J.** *Lesson Learned Preparing Process Implementation Descriptors (PIIDs)*. s.l. : NDIA Conference, 2004.
5. **Object Management Group.** *UML 2.0 OCL Specification*. 2003.
6. **Jonathan E. Cook, Alexander L. Wolf.** *Software process validation: quantitatively measuring the correspondence of a process to a model*. s.l. : ACM Transactions on Software Engineering and Methodology, Volume 8 Issue 2, April 1999, 1999.
7. *Software & Systems Process Engineering Meta-Model Specification Version 2.0*. Standard document URL: <http://www.omg.org/spec/SPEM/2.0/PDF> : Object Management Group, 2008. OMG Document Number: formal/2008-04-0.
8. **MSF Team, Microsoft.** *MSF Process Model v. 3.1*. For more information on MSF, see: <http://www.microsoft.com/msf> : Microsoft, 2002.
9. **IBM.** Rational Team Concert. [Online] Rational Software. [Cited: 04 29, 2010.] <http://www.ibm.com/developerworks/rational/products/rtc/>.

10. **Haumer, Peter.** *Eclipse Process Framework Composer. Delta Version: From the IBM donation to Version 1.x.* 3 10, 2006.
11. —. Eclipse Process Framework Composer - Part 1Key Concepts. [Online] 04 2007. [Cited: 04 29, 2010.] <http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf>.
12. *Un Método Heurístico para el Análisis y Selección de Herramientas de Modelado de Procesos de Desarrollo de Software.* **Silclir, Mauricio, et al.** Córdoba : s.n., 2010. JAIIO 2010. p. 14.
13. **Nien-Lin Hsueh, Wen-Hsiang Shen, Zhi-Wei Yang, Don-Lin Yang.** *Applying UML and software simulation for process definition, verification, and validation.* s.l. : Information and Software Technology, Volume 50 Issue 9-10, 2008.
14. **Object Management Group.** *UML 2.0 Specification.* 2003.
15. **P. Szyrko, M. Silclir, G. García Favre, D. Rubio.** *Un modelo de validación automático para la definición y mantenimiento de procesos de desarrollo de software.* San Juan : Workshop de Investigadores en Ciencias de la Computación WICC09, 2009.
16. **J.L.Pfleeger.** *Ingeniería del Software: Teoría y Práctica.* Buenos Aires : Prentice Hall, 2002. ISBN: 8131720985.
17. **WC3.** *Extensible Markup Language (XML).* 1996.
18. **SCAMPI Upgrade Team.** *Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM) A, Version 1.2: Method Definition Document.* Pittsburgh, Pennsylvania, USA : Software Engineering Institute, August 2006. HANDBOOK CMU/SEI-2006-HB-002.
19. **Leoncuk, C.** *Buenas prácticas técnicas que certifiquen una empresa TI para exportar estudio de un caso aplicando CMMI.* 2007.