

UN MÉTODO INCREMENTAL PARA EL ANÁLISIS VISUAL DE MODELOS DE PROCESO SOFTWARE

AN INCREMENTAL METHOD FOR VISUAL ANALYSIS OF SOFTWARE PROCESS MODELS



AUTOR

MARTA CECILIA CAMACHO O
Magister en Computación
*I. U. Colegio Mayor del Cauca
Docente Asociado
Grupo de Investigación I+D en
Informática
Grupo de Investigación IDIS**
cecamacho@unimayor.edu.co
COLOMBIA

*INSTITUCIÓN

Institución Universitaria Colegio
Mayor del Cauca
UNIMAYOR
Educación Superior
Popayán
Carrera 5 # 5-40
admision@unimayor.edu.co
COLOMBIA

AUTOR

JULIO ARIEL HURTADO ALEGRIA
Doctor en Ciencias de la Computación
**Unicauca
Docente Titular
Grupo de Investigación IDIS
ahurtado@unicauca.edu.co
COLOMBIA

**INSTITUCIÓN

Universidad del Cauca
UNICAUCA
Educación Superior
Popayán
Calle 5 No. 4 – 70
rectoria@unicauca.edu.co
COLOMBIA

AUTOR

PABLO HERNANDO RUIZ MELENJE
Magister en Computación
***Unicomfauca
Docente
Grupo de investigación Tic
Unicomfauca
Grupo de Investigación IDIS**
pruiz@unicomfauca.edu.co
COLOMBIA

***INSTITUCIÓN

Corporación Universitaria
Comfauca
UNICOMFAUCA
Educación Superior
Popayán
Calle 4 No. 8 – 30
rectoria@unicomfauca.edu.co
COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: El presente artículo surge como resultado de la ejecución de proyectos de especificación de diferentes procesos de desarrollo de software al interior de los grupos de investigación: IDIS, de la Universidad del Cauca, del grupo Investigación y Desarrollo en informática, de la Institución Universitaria Colegio Mayor del Cauca y del Grupo de investigación Tic Unicomfauca.

RECEPCIÓN: Octubre 16 de 2016

ACEPTACIÓN: Diciembre 10 de 2016

TEMÁTICA: Ingeniería del Software, Ingeniería de Procesos

TIPO DE ARTÍCULO: Artículo de Investigación Científica e Innovación

Forma de citar: Camacho. M. C. (2016). Un Método Incremental para el Análisis visual de modelos de Proceso software. En R, Llamasa Villalba (Ed.). Revista Gerencia Tecnológica Informática, 15(43), 79-91. ISSN 1657-8236.

RESUMEN ANALÍTICO

Especificar modelos de procesos en SPEM 2.0 es una tarea costosa por la cantidad de detalles a ser definidos. Normalmente dicha especificación evoluciona en la labor de definición del proceso o un proyecto de mejoramiento. En el desarrollo de software se pueden construir y evaluar entregables, de la misma forma se esperaría que con la definición del proceso ocurra igual. Sin embargo, la etapa de evaluación práctica de un proceso es mucho más larga, costosa y compleja que la misma etapa de definición. En este artículo se propone el uso de AVISPA-Method (Método incremental para el análisis visual de modelos de proceso software), como un método liviano para realizar evaluaciones tempranas del modelo de proceso a un costo mucho más reducido que su evaluación en la aplicación real, este método emplea a AVISPA (Analysis and Visualization for Software Process Assessment) como herramienta para la evaluación del modelo de proceso. Para mostrar la aplicabilidad del método, se ha desarrollado un estudio de caso, en el que el modelo de proceso Small-SPL se ha evaluado y depurado incrementalmente siguiendo este enfoque. Durante su proceso de modelado, AVISPA-Method, fue utilizado al terminar cada una de las versiones planificadas durante su especificación. Esto contribuyó a depurar la especificación de cada una de las tres versiones y a un notable mejoramiento en la especificación del proceso.

PALABRAS CLAVES: Vistas de Modelos de Procesos Software, AVISPA, Procesos Software, Modelos de Procesos Software, Verificación de Procesos Software, Análisis de Procesos Software.

ANALYTICAL SUMMARY

Specifying process models in SPEM 2.0 is a costly task by the number of details to be defined. Normally this specification evolves in the task of defining the process or an improvement project. In software development can be constructed and evaluated deliverables, in the same way, would expect that with the definition of the process occur the same. However, the practical evaluation stage of a process is much longer, costly and complex than the same stage of definition. This paper proposes the use of AVISPA-Method (Incremental method for visual analysis of software process models), a process visual analysis tool, as a lightweight method for conducting early evaluations of the process model at a much lower cost. To evidence the applicability of the method, a case study has been developed, in which the Small-SPL process model has been evaluated and incrementally debugged following this approach. During its modeling process, AVISPA-Method was used at the end each of the planned versions during its specification. This contributed to debug the specification of each of the three versions and a notable improvement in the specification of the process.

KEYWORDS: Software Process Model Blueprints, AVISPA, Software Process, Software process models, software process verification, software process analysis.

INTRODUCCIÓN

Un proceso software es el conjunto de herramientas, métodos y prácticas para producir un artefacto software [1] [2] [3] y un modelo de procesos es una representación abstracta de una arquitectura de proceso, de un diseño o definición para el entendimiento, toma de decisiones y representaciones de los distintos procesos a través de un ciclo de vida [4]. En la industria de software, los procesos son un factor determinante en la productividad de los proyectos y la calidad de los productos de software [5]. Por lo anterior, algunas organizaciones internacionales se han esforzado por definir estándares como ISO/IEC15504 [6], ISO/IEC12207 [7] y modelos

de madurez como el CMMI [8], con el fin de servir de procesos de referencia en el marco de proyectos de mejoramiento de los procesos de software en las organizaciones. Esto ha dado lugar a que cada vez se emplee una mayor cantidad de tiempo a la ingeniería y gestión del proceso de software, en la que una parte necesaria es evaluar y predecir la calidad del proceso mismo [9]. Debido al esfuerzo que las organizaciones empiezan a dedicarle a los procesos, la calidad de los modelos de proceso se convierte en un tema relevante para justificar su inversión. Independiente del modelo de calidad o estándar que se implemente, dado el costo de su definición y del impacto que tienen sobre la producción de software, es necesario evaluar aspectos

del proceso que permitan conocer si está correctamente especificado [10].

Para el análisis de los modelos de procesos existen varias alternativas no excluyentes, que incluyen su aplicación directa en proyectos piloto [11], su análisis a través de simulaciones [12], las métricas [13], las verificaciones formales [14] y la visualización [15].

La evaluación de los modelos de procesos software para determinar errores en su especificación es crucial para lograr la adopción satisfactoria por parte de las empresas. Pero la evaluación de los procesos software es una actividad compleja porque en muchos casos se realiza sobre el desarrollo de los proyectos, es decir cuando el proceso está desplegado en la organización y listo para su ejecución, donde las empresas sin saberlo pueden estar utilizando un proceso inadecuado, con errores comunes como tareas y actividades sobredimensionadas o subdimensionadas, tareas multipropósito y roles sobrecargados, lo cual puede implicar que los proyectos de la organización utilicen inadecuadamente los recursos asignados para el desarrollo de los proyectos y por lo tanto no se obtengan los resultados esperados. En el desarrollo de software se pueden construir y evaluar entregables, de la misma forma se esperaría que con la definición del proceso ocurra igual. Sin embargo, la etapa de evaluación práctica de un proceso es mucho más larga, costosa y compleja que la misma etapa de definición.

La visualización de modelos de proceso es una alternativa, para lograr en forma temprana, viable y efectiva, el análisis de modelos de procesos. Una primera aproximación son los Blueprints propuestos por Hurtado y otros [16], los cuales son representaciones gráficas de modelos de proceso de software que permiten a los diseñadores de procesos: (i) mejorar la calidad de los modelos de procesos de software, (ii) identificar las anomalías o problemas en la especificación y/o el modelo, (iii) proporcionar pistas para encontrar inconsistencias en el modelo. Como siguiente aproximación se tiene los patrones de errores de AVISPA [10], los cuales permiten identificar de manera concreta problemas específicos en el modelo de proceso, resaltándolos y diferenciando los elementos con algún tipo de anomalía.

La visualización ha mostrado su efectividad en el análisis de modelos de procesos [10] [17], sin embargo se carece de guía para que los ingenieros de proceso aprovechen sus bondades. La idea es que un ingeniero de procesos se pueda beneficiar de un ciclo completo PDCA (Plan, Do, Check, Act) en la depuración del modelo de procesos de software. Es decir, dado que el modelo de proceso evoluciona y se define incrementalmente, en este trabajo se plantea que la verificación de la calidad

del modelo pueda ser parte del ciclo justo antes del hacer (Do) y después de tener un nuevo diseño del modelo de procesos (Plan) resultado de las mejoras analizadas a partir del chequeo (Check) y propuestas en el actuar

En este artículo se presenta mediante un estudio de caso, como AVISPA-Method fue utilizado para mejorar incrementalmente la calidad del modelo de referencia Small-SPL, un modelo orientado a la adopción del enfoque de líneas de productos en pequeñas organizaciones a través de la reutilización de activos de proceso.

El resto de este artículo se organiza de la siguiente manera: la sección I presenta trabajos relacionados que abordan el problema del análisis de la calidad de los modelos de proceso, la sección II presenta una síntesis del trabajo con la herramienta AVISPA. En la sección III se describe el enfoque incremental para el análisis de modelos de proceso AVISPA-Method. La sección IV presenta el estudio de caso en el que Small-SPL es analizado utilizando AVISPA-Method. La sección V presenta algunas conclusiones, limitaciones y trabajo futuro.

1. TRABAJOS RELACIONADOS

El interés de este trabajo está en poder evaluar la calidad de los modelos de procesos especificados en SPEM 2.0. A continuación presentamos algunos trabajos que se enfocan en la evaluación de la calidad del proceso o de su especiación.

En la ingeniería de requisitos se utilizan varias técnicas para realizar la especificación correcta y precisa de los requisitos. A menudo se crean procesos/herramientas que combinan múltiples técnicas donde la salida de una técnica se convierte en la entrada a la siguiente. En [18] se modela y se estudia un proceso requisitos para comprobar la coherencia de los requisitos mediante la evaluación de errores. Se realiza un análisis de los factores de entrada para determinar el efecto que las fuentes de incertidumbre pueden tener sobre la precisión final del proceso de verificación y de la coherencia de los requisitos.

En un enfoque de evaluación basado en evidencias se toman los resultados o evidencias de la ejecución del modelo aplicado a un proyecto, y se comparan con las especificaciones de algún modelo de referencia [11]. Un ejemplo de evaluación de procesos basada en evidencia ocurre durante la verificación del cumplimiento respecto a algún modelo de referencia como CMMI [8] e ISO/IEC15504 [6]. Normalmente este tipo de actividades de evaluación basada en evidencia sólo pueden llevarse a cabo una vez que el modelo de proceso se ha aplicado a proyectos específicos. Así, la evaluación

basada en evidencia del modelo de proceso, necesita ciclos muy largos. En este contexto los procesos no necesariamente debe especificarse en algún lenguaje formal o semiformal, debido a que se debe verificar el cumplimiento o adaptación al estándar, y aun así no se garantiza la calidad del modelo de proceso mismo.

Schramm et al. [19] describe que dentro de las actividades de la ingeniería de modelos de procesos se debe considerar la evaluación de la descripción del modelo de proceso para proporcionar información de mejora. La evaluación del proceso se hace mediante la recolección de conceptos, herramientas y la medición de las características de las descripciones de proceso. La información recolectada puede ser utilizada para realizar una actualización de la descripción del proceso y para la extracción de información útil en la mejora del proceso. Ruiz et al. [20] propone un meta-proceso, para formalizar el proceso software usando la plataforma Eclipse Process Framework. El meta proceso define una fase de ejecución donde las actividades, *Partial Validation* y *Complete validation*, son las encargadas de efectuar la evaluación del proceso software. Con el desarrollo de estas actividades se valida la correctitud de la especificación del proceso.

Gruhn et al. [14] plantea una técnica de verificación basada en los resultados de la traza de la ejecución de los procesos. Si para varias ejecuciones el proceso demuestra funcionar bien (p ej. ausencia de cuellos de botella), entonces es muy probable que el modelo de procesos que lo define este bien elaborado. Pero no se puede asegurar que será así para todas las ejecuciones. Este enfoque requiere de varias ejecuciones del modelo lo que lo hace lento. Por lo tanto, la verificación de modelos de proceso de software ayuda a prevenir la ejecución de errores en los modelos de proceso de software.

Un enfoque de evaluación de la usabilidad y mantenibilidad del modelo de proceso a través de métricas del modelo, es propuesto por Cánfora et. al. [13]. Los autores definen un conjunto de métricas para medir de manera global la usabilidad y mantenibilidad de un modelo de proceso de software, basados en un conjunto de métricas aplicadas a todo el modelo. La propuesta permite evaluar aspectos generales del modelo de proceso tales como número de tareas por rol, número de productos de trabajo de entrada en promedio, entre otros. En este caso, es difícil conocer en forma específica los problemas del modelo de proceso, así como la ubicación de los mismos dentro del modelo. Esto le resta efectividad al momento de definir oportunidades de mejora.

La simulación del modelo de proceso es otra alternativa para analizar algunos aspectos del modelo de proceso

[21] [22]. En éste caso se tiene un ciclo de revisión más corto que la ejecución misma del proceso, pero aún requiere de gran cantidad de datos históricos. Esta técnica de análisis es adecuada sólo cuando se sabe que el modelo de proceso es conveniente para la organización o el proyecto, pero si el modelo está incompleto, insuficientemente especificado o no es conveniente, la simulación no producirá los resultados esperados.

Dado que el análisis de modelos de software ayuda a disminuir los recursos, esfuerzo y de la misma forma recursos hardware, Christov y Avrunin [23], proponen una técnica formal de verificación y validación para identificar y medir las diferencias entre los modelos de procesos y las ejecuciones reales de los procesos. Sin embargo, este análisis está orientado a evaluar la congruencia entre el modelo y su ejecución en el proyecto. Una de las dificultades para la obtención de estas mediciones, es que su realización no es fácil y los resultados no son precisos [24].

Hurtado et al. [16] propone un enfoque visual para la evaluación de modelos de procesos. La evaluación se basa en tres tipos de vistas arquitectónicas, enfocadas en elementos básicos de un proceso software: Role Blueprint, Task Blueprint y Work Product Blueprint. Este acercamiento utiliza la herramienta ProcessModel, basada en Modrian and Mose, para la visualización de los modelos de procesos. Las diferentes visualizaciones del proceso que se obtienen al usar la herramienta son utilizadas para encontrar errores, problemas y mejoras en la especificación del proceso. Hurtado et al. [25] define una serie de patrones de error que indican la presencia potencial de conceptos erróneos en modelos de procesos software. En este trabajo se presentan patrones de error, se caracterizan los tipos de errores y se detalla cómo los errores pueden ser localizados en un modelo de proceso de software. Para ayudar en el análisis de la calidad de la especificación de los procesos se propone AVISPA como una herramienta que representa gráficamente diferentes aspectos de un modelo de proceso y resalta posibles errores como indicadores intuitivos y comprensibles. Hurtado et al. [26] describe AVISPA para el análisis de la calidad de los modelos de procesos de software definidos en SPEM 2.0. Se proporciona una descripción detallada de los elementos internos de AVISPA para mostrar tanto su estructura como sus mecanismos de extensibilidad. También presentan un mecanismo interactivo para definir nuevos scripts de análisis e implementar nuevos patrones y vistas del proceso.

El enfoque de AVISPA permite analizar los modelos de proceso en forma temprana, aún si los modelos son incompletos, aprovechando el potencial de la

visualización para facilitar la identificación y localización de los puntos problemáticos del proceso en una forma más intuitiva [16] [10] [26]. Sin embargo, los reportes de utilización de AVISPA muestran muy poco la forma en que el análisis puede ayudar a la especificación del modelo de proceso, limitándose a presentar los resultados del análisis y la efectividad del enfoque a través de distintos estudios de caso, si definir la forma en que el análisis se puede dar, ni su relación con la actividad de formulación.

En este trabajo se define el método AVISPA-Method para guiar el análisis de modelos de proceso con la herramienta AVISPA, siguiendo un enfoque incremental sincronizado con la formalización del modelo de procesos, el método es evaluado a través de un estudio de caso donde se verifica la correctitud del modelo Small-SPL.

TABLA 1. Patrones de error identificados por AVISPA

Patrón de error	Descripción	Localización	Identificación
Rol Aislado	Rol que no colabora con otros o no ha sido asignado	Vista de roles	Nodo no conectado
Subproyectos independientes	Subgrafos Aislados	Vista de Tareas Vista de Productos de Trabajo	Grupos de nodos independientes.
Rol sobrecargado	Un rol con excesivas tareas asignadas	Vista de roles	Nodos con una desviación superior a la media
Tarea multipropósito	Una tarea con propósito múltiple	Vista de tareas	Ancho del nodo mayor a la media
Producto de trabajo basura	Un producto de trabajo no utilizado por ninguna tarea	Vista de productos de trabajo	Nodo aislado
Producto de trabajo sobredemandado	Un producto de trabajo utilizado por muchas tareas	Vista de productos de trabajo	Nodos con una desviación superior a la media
elemento sin guía asociada	Un elemento sin guía asignada	En las tres vistas	un nodo con color azul

Fuente: [18]

3. METODO INCREMENTAL PARA EL ANALISIS VISUAL DE MODELOS DE PROCESO SOFTWARE AVISPA-METHOD

AVISPA-Method considera las siguientes actividades como se muestra en la figura 1:

2. AVISPA

AVISPA es una herramienta de evaluación gráfica, la cual toma como entrada un modelo de proceso especificado en SPEM 2.0 exportado como un archivo XML desde Eclipse Process Framework Composer; AVISPA aplica sobre el modelo de proceso una serie de métricas de software con las que genera tres vistas: Vista de Roles, Vista de Tareas y Vista de Productos de Trabajo, cada una enfocada a un elemento básico empleado en la definición de procesos [10]. En cada una de estas representaciones visuales, AVISPA identifica anomalías y las resalta sobre los elementos del proceso [26] utilizando un conjunto de patrones de error [10] listados en la tabla 1. El objetivo de los patrones de error es ayudar a los ingenieros de procesos en la evaluación de la calidad, identificando errores recurrentes en las especificaciones de un proceso de software y planteando como buscar y analizar su corrección.

1. Diseñar el modelo de procesos: el ingeniero de procesos diseña y formaliza una versión del modelo de procesos (versión SPEM).
2. Exportar el modelo de procesos: el ingeniero de procesos realiza la exportación del modelo de procesos actual (versión SPEM) a una versión XML,

donde esta última versión es compartida al analista de procesos

3. Examinar el proceso con AVISPA-Method: el analista de procesos realiza la carga del modelo de proceso (versión XML) en la herramienta AVISPA. Sobre cada una de las vistas y con la ayuda de cada uno de los patrones de error, identifica y localiza problemas potenciales y oportunidades de mejora del modelo. Si no se identifican problemas, en la definición del modelo de procesos, se puede dar por terminado el análisis.
4. Análisis y reporte de resultados: el analista de procesos y el ingeniero de procesos se reúnen para revisar y discutir los problemas identificados en el modelo de procesos. Este análisis requiere revisar el modelo de procesos en su formato original (versión SPEM) y eventualmente ir con los participantes del proceso a evaluar aspectos prácticos de su aplicación. Al final de la revisión se identifican los problemas reales sobre el modelo de proceso, así como las posibles mejoras a realizar. El analista reporta los errores y sugerencias de mejora y se la entrega al ingeniero de procesos.
5. Realizar ajustes: el ingeniero de proceso ajusta el modelo de procesos de acuerdo a los errores y a las sugerencias de mejora identificadas. Si no es la última versión o el modelo de procesos aun contiene errores significativos, entonces se debe ir al primer paso.

Los roles y artefactos que hacen parte de AVISPA-Method se muestran a continuación:

Roles:

- Ingeniero de procesos: Lidera y coordina el diseño y la formalización de los modelos de procesos software
- Analista de procesos: Identifica y localiza problemas potenciales y oportunidades de mejora del modelo de procesos

Artefactos:

- Modelo de procesos SPEM: archivo digital de la versión del modelo de procesos especificado en SPEM
- Modelo de procesos XML: archivo digital del modelo de procesos versión XML, el cual es el insumo principal para el análisis
- Reporte de errores: Documento que plasma el reporte de errores y las oportunidades de mejora identificadas en el análisis del proceso

4. ESTUDIO DE CASO: SMALL-SPL

4.1 CONTEXTO DEL ESTUDIO DE CASO

Small-SPL es un modelo de proceso de desarrollo que busca mediar entre las exigencias de producción de líneas de productos y las características de las pequeñas y medianas entidades desarrolladoras de software. El ciclo de vida de Small-SPL incluye tres subprocesos: Ingeniería de Dominio - ID e Ingeniería de Producto - IP, engranados por un tercer subproceso denominado Gestión de Activos basado en Requisitos - GAR, como puede observarse en la figura 2 [27]. Para la especificación del proceso Small-SPL se utilizó SPEM 2.0 [28], como meta- modelo y lenguaje de modelado de procesos, esto se hizo mediante la plataforma Eclipse Process framework Composer-EPFC.

FIGURA 1. Método de análisis incremental para el análisis modelos de proceso con AVISPA-Method.

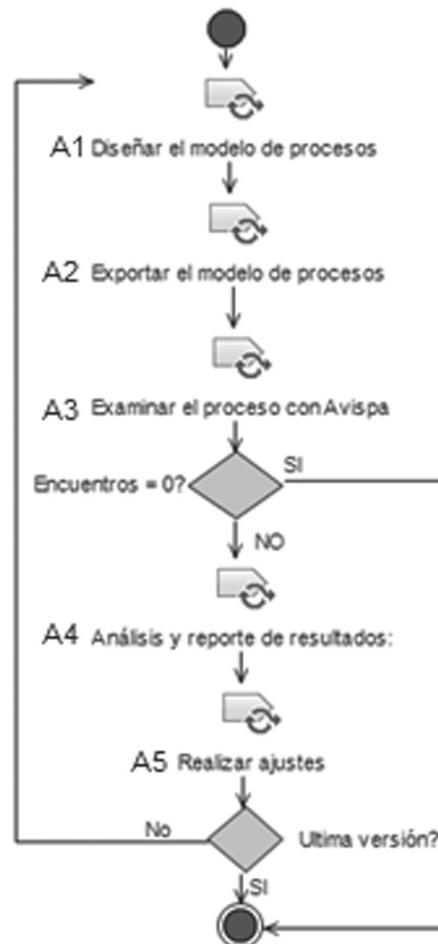
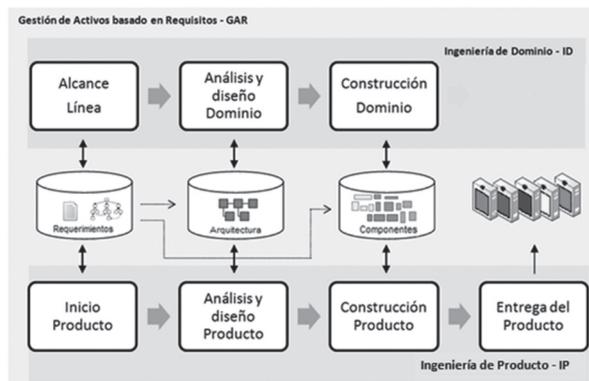


FIGURA 2. Proceso Small-SPL [27]

Especificar un proceso software es una tarea intensiva en conocimiento y por tanto propensa a errores amenazando su calidad y su correcto uso en proyectos y organizaciones. Al proponer un proceso de desarrollo, como es el caso de Small-SPL, es necesario evaluar la correctitud de su especificación para lo cual se siguió AVISPA-Method.

4.2 DISEÑO DEL ESTUDIO DE CASO

Para validar Avispa-Method se desarrolló un estudio de caso, basado en la propuesta de Yin et al. [29] y Runeson et al [30]. Concretamente este estudio de caso examina que ocurre en la especificación de un proceso al aplicar el AVISPA-Method, el cual fue de tipo holístico descriptivo y cuya unidad de estudio fue el proceso Small-SPL. La pregunta definida para el del estudio de caso fue: *¿Usar AVISPA-Method sirve para depurar y mejorar la especificación del modelo de proceso?*

El objeto de este estudio de caso fue observar y analizar los aportes de AVISPA-Method como estrategia de análisis para asegurar la calidad del modelo de proceso Small-SPL antes de su despliegue. Teniendo en cuenta el objeto del estudio de caso, se definieron los indicadores, métricas e instrumentos presentados en la tabla 2.

Se denominan hallazgos a las alertas generadas por la herramienta AVISPA y errores a los hallazgos que se verifican como fallos reales de la especificación del proceso al ser analizada con AVISPA-Method.

4.3 DESARROLLO DEL ESTUDIO DE CASO

Para el desarrollo del estudio de caso se utilizó Avispa-Method. Se definieron incrementalmente tres diferentes versiones de Small SPL donde cada versión fue examinada con AVISPA y los resultados obtenidos fueron analizados.

- **Versión 1.0:** corresponde la primera definición del proceso la cual se realizó basada en los referentes teóricos estudiados, es decir, Framework para SPL del SEI [31], PuLSE de Instituto Fraunhofer [32] y FAMILIES del ESI [33] [34], y en un estudio y caracterización de las pequeñas empresas productoras de software [35].
- **Versión 2.0:** corresponde a la segunda definición del proceso obtenida después de aplicar la versión 1.0 en un caso de estudio experimental [17] y en la evaluación empresarial de Small-SPL [27].
- **Versión 3.0:** corresponde a la tercera especificación del proceso obtenida al realizar la evaluación de la aplicabilidad y utilidad del proceso Small-SPL [27].

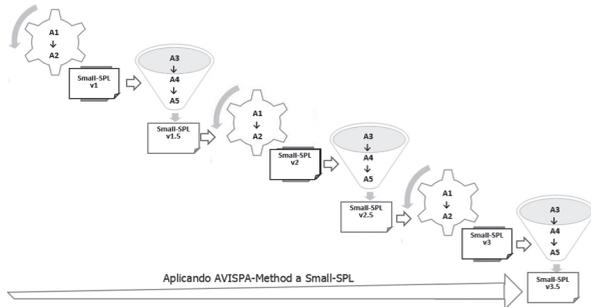
TABLA 2. Indicadores, Métricas e Instrumentos del estudio de caso

Indicadores	Métricas	Instrumentos
Utilidad de AVISPA-Method	<u>#hallazgos</u>	Vista de Roles
	#total elementos	Vista Productos de trabajo
	#errores #hallazgos	Vistas de Tareas Análisis del resultados
Incremento de la correctitud	# errores disminuidos entre versiones	Tabla comparativo de hallazgos

Cada una de las iteraciones tuvo un objetivo específico y contribuyó en la definición de diferentes versiones de Small-SPL, cada una de las versiones fue evaluada con AVISPA-Method, siguiendo cada una de sus actividades, para verificar la correctitud del modelo de proceso, es de aclarar que después de realizar la evaluación de cada una de las versiones de Small-SPL, se obtenía una versión mejorada después de corregir los errores encontrados a partir de los resultados de la evaluación, estas versiones se denominaron x.5, y era la base para la especificación de la siguiente versión. La aplicación de AVISPA-Method se grafica en la figura 3.

A continuación se realizará un análisis de los resultados obtenidos, los cuales se detallaran a partir de las tres vistas gráficas (tareas, roles y productos de trabajo) que proporciona AVISPA de un modelo de procesos, cada una trata un aspecto particular del modelo de procesos [16].

FIGURA 3. Aplicando AVISPA-Method en la especificación de Small-SPL

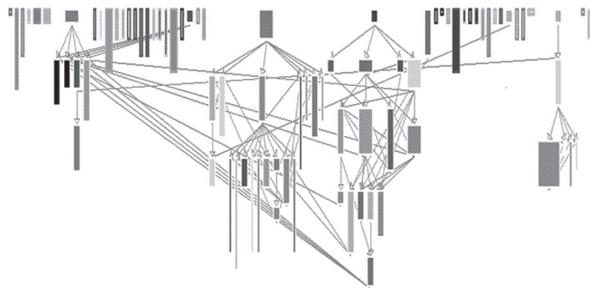


4.3.1 Hallazgos en el análisis de Small-SPL en la vista de tareas

La vista de tareas proporcionada por AVISPA muestra el proceso desde la perspectiva de las tareas efectuadas durante la ejecución del modelo de proceso. En esta vista, cada nodo rectangular representa una tarea específica del proceso y los atributos de cada nodo proporcionan información sobre el proceso que está siendo analizado [26]. Las figuras 3 a la 5 presentan los resultados obtenidos al evaluar las tres versiones de Small-SPL.

Los resultados obtenidos al evaluar la primera versión de Small-SPL con AVISPA-Method, evidenciaron varios errores en la especificación del modelo de proceso. Uno de los patrones de error definidos es subproyectos independientes que pueden observarse como nodos aislados, en la vista de tareas de la versión 1.0 (figura 4) puede observarse varias tareas desconectadas (rectángulos apilados en la parte superior de la figura), estas tareas no aportan valor al objetivo del proceso, actúan de manera apartada y fuera del resto del proceso. El porcentaje de tareas aisladas fue del 47% para esta primera versión. Para obtener la versión 1.5 se revisó los productos de trabajo que ingresan y salen de cada tarea.

FIGURA 4. Vista de Tareas Small-SPL v 1.0

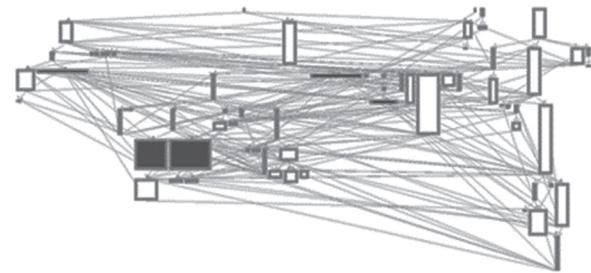


La figura 5 corresponde a la vista de tareas obtenida al analizar con AVISPA la versión 2.0 de Small-SPL. Lo

primero que puede observarse es que no se detectaron subproyectos independientes, no se encuentran nodos aislados.

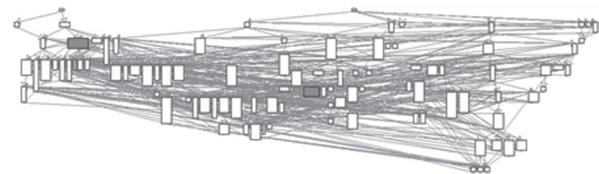
Otro de los patrones de error es tareas multipropósito correspondientes a tareas con más de un objetivo, se identifican visualmente como nodos más anchos que el promedio que se han coloreado más oscuros que los demás en la figura 5. Para cada una de las tareas multipropósito reportadas por AVISPA se examinó el objetivo de la tarea, los productos de trabajo originados, así como también cada uno de sus pasos, donde se encontró disparidad, para lo cual se recomendó dividir la tarea.

FIGURA 5. Vista Tareas Small-SPL v 2.0



El resultado del análisis de la última versión de Small-SPL para la vista de tareas se puede observar en la figura 6, en la que se evidencia primero que hay un incremento en el número de tareas del proceso, como también que no existen subproyectos aislados, se distinguen que persisten dos tareas resaltadas con un color más oscuro, sin embargo al examinar la especificación de la dos tareas resaltadas no se encontraron objetivos dispares o equivocaciones al enlazar los productos de trabajo resultantes; por lo tanto no se optó por dividir las tareas y se considera la recomendación de AVISPA como un falso positivo.

FIGURA 6. Vista de Tareas Small-SPL v 3.0

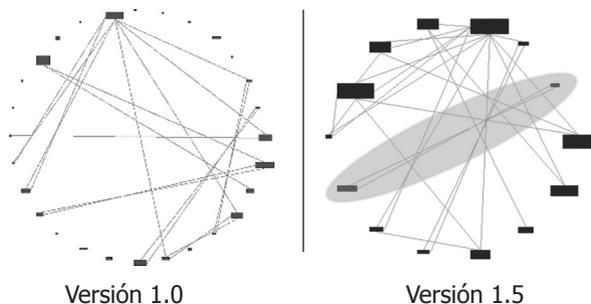


4.3.2 Hallazgos en el análisis de Small-SPL en la vista de Roles

Los roles del proceso corresponden al segundo elemento que se puede verificar con AVISPA en un modelo de proceso, en la vista de roles cada nodo identifica un rol y cada una de las líneas entre nodos especifica colaboración entre ellos [25]. Esta vista es muy

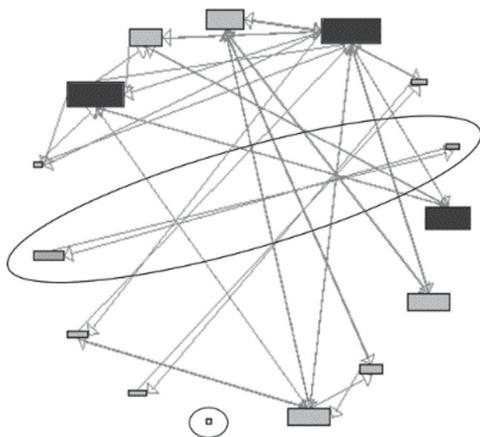
importante para el análisis de Small-SPL debido a que las pequeñas empresas productoras de software tienen limitaciones de personal.

FIGURA 7. Vista de Roles Small-SPL v 1.0



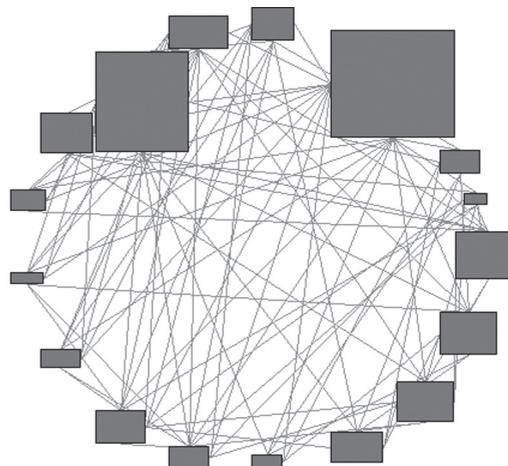
En la figura 7 se puede observar el análisis de la primera versión de Small-SPL con AVISPA, la parte izquierda de la figura 7 muestra los hallazgos de la primera revisión donde se visualizan muchos nodos solitarios lo que se reconoce como el patrón de error Rol aislado, este tipo de patrón de error muestra una especificación errónea dado que no suele ser correcto tener un rol que nunca colabore en ninguna tarea y con los otros roles [10]. Este hallazgo se consideró como un resultado preocupante, debido a que se detectó duplicidad de roles, una vez se depuró el proceso se llegó a los resultados reflejados en la parte derecha de la figura 7, en la que se visualiza que se disminuyó la cantidad de roles, como también puede observarse que existen dos equipos de trabajo, uno conformado por la mayoría de los roles, relacionados entre sí y otro compuesto solo por dos nodos (sombreados por una elipse) que corresponden al patrón de error Subproyectos independientes de AVISPA [10].

FIGURA 8. Vista de Roles Small-SPL v 2.0



Al realizar el análisis con AVISPA de la segunda versión de Small-SPL se obtuvo como resultado la vista de roles de la figura 8, en la que se puede observar un rol aislado, un subproceso independiente y de acuerdo al patrón rol sobrecargado existen tres nodos de mayor tamaño (más oscuros) que se indicaban como roles sobrecargados

FIGURA 9. Vista de Roles Small-SPL v 1.0



La figura 9 es el resultado de evaluar a Small SPL en su tercera versión, puede observarse que hay más conexiones entre nodos, lo que indica que la colaboración entre roles es mayor. Existen dos nodos sobresalientes, que indican que aún persisten dos roles sobrecargados, por lo cual debió revisarse la participación del rol en las diferentes tareas. Los nodos resaltados corresponden a los líderes de desarrollo de cada subproceso de Small-SPL (ingeniero de dominio e ingeniero de producto), por lo cual su participación en el desarrollo de tareas es mayor que la de otros roles.

4.3.3 Hallazgos en el análisis de Small-SPL en la vista de Productos de Trabajo

La vista de productos de trabajo proporcionada por AVISPA en la evaluación de procesos, permite a los ingenieros de proceso visualizar los productos de trabajo aislados, sobre-demandados y productos de trabajo innecesarios ("grasa"). En estas vistas cada nodo representa un producto de trabajo, su tamaño indica cuantas tareas escriben, leen o emplean ese producto de trabajo. La vista productos de trabajo ofrece una importante caracterización del modelo de proceso, permitiendo visualizar la carga y esfuerzo que indicaría para las pequeñas entidades acoger y seguir el proceso Small SPL.

En la figura 10 pueden observarse los resultados obtenidos al evaluar la versión 1.0 de Small-SPL, en la

parte superior se encuentran varios nodos aislados que indican que pueden ser productos de trabajo consumidos (requeridos) pero no generados. El patrón productos de trabajo innecesarios ("grasa") son los nodos oscuros que pueden indicar que estos productos no son consumidos por ninguna tarea.

Al evaluar la versión 2.0 de Small SPL se pudo evidenciar que si bien el número total de productos de trabajo no disminuyó de manera considerable, los ajustes realizados lograron reducir los productos de trabajo desconectados (no son producidos por ninguna tarea, pero si consumidos) y también se disminuyeron los productos de trabajo "grasa" que son generados, pero no consumidos, estos hallazgos pueden observarse en la figura 11.

La figura 12 es el resultado de evaluar a Small-SPL en su versión 3.0, puede observarse que las conexiones entre nodos son mayores, lo que indica una mayor integración entre los productos de trabajo, no existen nodos desconectados, pero si se observan tres productos de trabajo basura o grasa, los cuales al revisar la especificación de proceso eran descritos como entradas opcionales.

FIGURA 10. Vista Productos de trabajo Small-SPL v 1.0

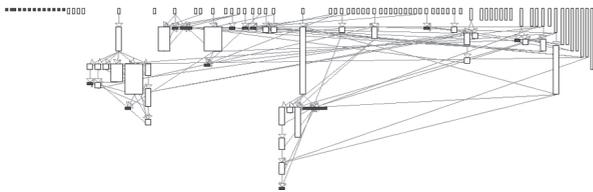


FIGURA 11. Vista de Productos de trabajo Small-SPL v 2.0

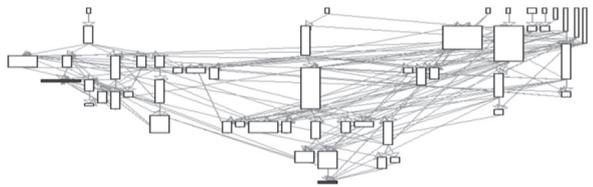
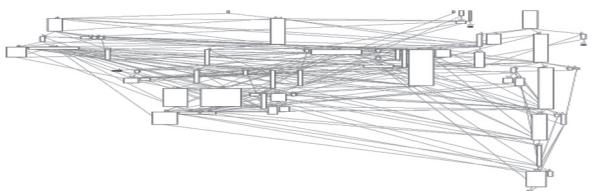


FIGURA 12. Vista de Productos de trabajo Small-SPL v 3.0



4.3.4 Comparativo de hallazgos en el análisis de Small-SPL

El comparativo de los resultados obtenidos para cada versión permite observar los errores detectados, los cambios entre versiones y las depuraciones realizadas a la especificación del proceso.

FIGURA 13. Resultados comparativos del análisis de Small-SPL con AVISPA-Method

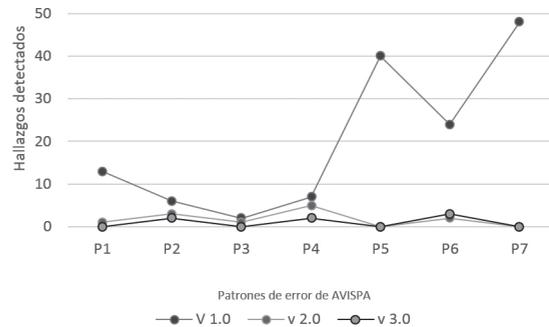


TABLA 3. Comparativo de hallazgos en las versiones de Small-SPL con AVISPA-Method

	Patrón de error	Small SPL 1.0	Small SPL 2.0	Small SPL 3.0
P1	Rol Aislado	13	1	0
P2	Rol sobrecargado	6	3	2
P3	Subproyectos independientes (roles)	2	1	0
P4	Tarea multipropósito	7	5	2
P5	Subproyectos independientes (tareas)	40	0	0
P6	Producto de trabajo basura	24	2	3
P7	Subproyectos independientes (productos de trabajo)	48	0	0

En la figura 13 y en la tabla 3 se puede apreciar un comparativo de los resultados obtenidos al utilizar AVISPA-Method en las diferentes versiones de Small SPL. Se puede observar, que la mayor cantidad de errores se encontraron en la primera versión del

proceso, en la segunda versión el número de hallazgos disminuyó considerablemente y en la tercera versión se logró eliminar casi por completo los errores de la especificación

4.4 RESULTADOS DEL ESTUDIO DE CASO

Una vez realizado el estudio de caso sobre AVISPA-Method se analizaron los resultados obtenidos por medio de los indicadores e instrumentos previamente definidos, a continuación, se presentan estos resultados.

a) Indicador utilidad

Como se puede observar en la figura 16, y en las tablas 3 y 4, la aplicación de AVISPA-Method permitió obtener versiones que evolucionaron en la correctitud de su especificación. También se puede observar en la segunda fila de la tabla 4 que los hallazgos reportados por AVISPA-Method tienen un gran valor de asertividad en cuanto a identificar errores reales.

TABLA 4. Promedio de indicadores de utilidad

	Small SPL 1.0	Small SPL 2.0	Small SPL 3.0
Promedio			
$\frac{\# \text{hallazgos}}{\# \text{total elementos}}$	28,5%	6,4%	2,4%
Promedio			
$\frac{\# \text{errores}}{\# \text{hallazgos}}$	1	0.94	0.95

b) Indicador Incremento de la correctitud

En la figura 13 y en la tabla 3 se puede observar el número de errores disminuidos entre versiones. Si se compara los resultados obtenidos al evaluar la primera versión de Small-SPL con los resultados de la segunda, la diferencia es considerable, para la primera versión se identifican 140 hallazgos que advierten de posibles errores en la especificación del proceso, mientras en la segunda versión se encuentran 12 hallazgos, una disminución del 91%. Entre la segunda y la tercera versión del proceso la diferencia no es tan notable, de 12 hallazgos se pasa a 7 en la tercera versión.

De acuerdo a los resultados obtenidos en este estudio de caso se considera que aplicar AVISPA-Method ayuda a depurar y mejorar la especificación de procesos, además

por ser un proceso incremental facilita el análisis y la corrección de las fallas de especificación, asegurando un refinamiento continuo y ayudando a que la correctitud del modelo de proceso sea mayor.

AVISPA-Method ejercita al ingeniero de procesos de tal forma que le permite adquirir habilidades y conocimientos que harán que sus especificaciones sean más correctas.

5. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presenta un método incremental para el análisis visual de modelos de proceso software AVISPA-Method, cuyo objetivo es ayudar en la evaluación temprana de modelos de procesos de forma tal que sea de bajo costo con respecto a su evaluación en la aplicación real. De acuerdo con los resultados obtenidos en el estudio de caso, AVISPA-Method ayudó en la depuración, optimización y evolución de la especificación del proceso Small-SPL en cada una de sus versiones, permitiendo que el ingeniero de procesos pueda elevar la correctitud de la especificación de forma incremental mediante un refinamiento continuo.

El ingeniero de procesos al conocer los errores típicos en la especificación del proceso puede reducir el número de errores identificados y el tiempo de localización de la causa mejorando el modelado y la especificación del proceso. La verificación de correctitud del proceso permite que la especificación del proceso sea verificada antes de que este sea desplegado, lo cual es de gran ayuda para las empresas debido a que la evaluación práctica de un proceso es mucho más larga, costosa, riesgosa y compleja que la misma etapa de definición.

Es importante resaltar que el modelo de proceso se valida a través de estudios de caso y que AVISPA actúa como un complemento en la obtención de la calidad en la especificación del modelo de proceso. Como trabajo futuro se espera aplicar AVISPA-Method, en la definición y formalización de nuevos procesos, de tal forma que cada experimentación permita su mejoramiento.

6. AGRADECIMIENTOS

Este trabajo fue financiado parcialmente por el proyecto red de formación de talento humano para la innovación social y productiva en el departamento del Cauca - INNOVACIÓN CAUCA ejecutado por la Universidad del Cauca bajo el código 3848

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] W. S. Humphrey, «Managing the Software Process,» Addison-Wesley Longman, 1989.
- [2] S. Hossein y C. Natsu, «Characterizing a Software Process Maturity Model for Small,» 1997.
- [3] M. Ginsberg y L. Quinn, «Process Tailoring and the software Capability Maturity Model.,» 1994.
- [4] P. Feiler y H. W., «Software Process Development and Enactment: Concepts and Definitions,» *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-92-TR-004,* 1992.
- [5] A. Fuggetta, «Software Process: A Roadmap,» de *International Conference in Software Engineering - Future of Software Engineering Track,* 2000.
- [6] ISO, «ISO/IEC 15504 : Information Technology - Software Process,» 1998.
- [7] ISO/IEC, «ISO/IEC 12207:2008 Systems and Software Engineering,» 2008.
- [8] SEI, «CMMI for Development, Version 1.2,» 2006.
- [9] B. Cheng, R. d. Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, A. Becker, N. Bencomo, Y. Brun, B. Cukic, G. D. Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. M. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns y J. Whittle, «Software Engineering for Self-Adaptive Systems,» de *Software Engineering for Self-Adaptive Systems,* vol. 5525, S. B. Heidelberg, Ed., Berlin, Springer, 2009.
- [10] J. A. Hurtado, M. C. Bastarrica y A. Bergel, «Analyzing software process models with AVISPA,» de *ICSSP '11 Proceedings of the 2011 International Conference on Software and Systems Process,* 2011.
- [11] L. J. Osterweil, «Software Processes are Software Too,» de *9th International Conference on Software Engineering ICSE,* Los Alamitos, CA, USA, 1987.
- [12] S. Sutton, «Advancing process modeling, simulation, and analytics in practice,» de *ICSSP,* Zurich, 2012.
- [13] G. Canfora, F. García, M. Piattini, F. Ruiz y C. A. Visaggio, «A family of experiments to validate metrics for software process models,» de *Journal of Systems and Software,* 2005.
- [14] V. Gruhn, «Validation and verification of software process models,» *Software Development Environments and CASE Technology,* pp. 271-286, 1991.
- [15] J. Ge, H. Hu, Q. Gu y J. Lu, «Modeling Multi-View Software Process with Object Petri Nets,» de *International Conference on Software Engineering Advances ICSEA,* Tahiti, 2006.
- [16] J. A. Hurtado Alegría, A. Lagos, A. Bergel y M. C. Bastarrica, «Software process model blueprints,» de *Proceedings of the 2010 international conference on New modeling concepts for today's software processes: software process,* Padernorn, Germany, 2010.
- [17] M. C. Camacho, J. A. Hurtado y P. Ruiz, «Experiencias de Desarrollo de Líneas de Productos Software en el Contexto de un Curso de Ingeniería de Software,» *Revista I + T + C: Investigación, Tecnología y Ciencia,* vol. 6, 2014.
- [18] L. Wenbin, J. H. Hayes, A. Giulio y G. Yann-Gaël, «Error leakage and wasted time: sensitivity and effort analysis of a requirements consistency checking process,» *Software evolution and Process,* 2016.
- [19] J. Schramm, P. Dohrmann, A. Rausch y T. Ternité, «Process Model Engineering Lifecycle: Holistic Concept Proposal and Systematic Literature Review,» *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on,* 2014.
- [20] P. Ruiz y J. Hurtado, «Formalizing the Software Process in Small Companies,» *Conferencia Colombiana de Computación,* 2012.
- [21] W. Scacchi, «Understanding Software Process Redesign using Modeling, Analysis and Simulation,» *Software Process Improvement and Practice,* vol. 5, p. 183-195, 2000.
- [22] M. Ruiz Carreira, I. Ramos Román y M. Toro Bonilla, «Marco dinámico integrado para la mejora de los procesos software».
- [23] S. Christov, G. S. Avrunin y L. A. Clarke, «Using Event Streams to Validate Process Definitions,» Amherst, 2009.
- [24] K. El Emam, «A methodology for validating software product metrics,» 2002.
- [25] J. A. Hurtado Alegría, M. C. Bastarrica y A. Bergel, «AVISPA: a tool for analyzing software process models,» *Journal of software: Evolution and Process,* vol. 26, n° 4, p. 434-450, 2014.
- [26] J. A. Hurtado Alegría, M. C. Bastarrica y A. Bergel, «AVISPA: A Tool for Analyzing Software Process Models,» de *JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS,* 2013.
- [27] M. C. Camacho y J. A. Hurtado, «SPL en las PYMES desarrolladoras de software del cauca: una experiencia desde Colmayor,» Universidad del Cauca, 2013.
- [28] OMG, «Software Process Engineering Metamodel SPEM 2.0,» 2008.
- [29] R. K. Yin, Case study research. Design and methods, 3, Ed., London, Thousand Oaks: Sage Publications, 2003.

- [30] P. Runeson y M. Höst, «Guidelines for conducting and reporting case study research in software engineering,» *Empirical software engineering*, vol. 14, n° 2, pp. 131-164, 2009.
- [31] L. Northrop, P. Clements, F. Bachmann, J. Bergey, G. Chastek y S. Cohen, «A Framework for Software Product Line Practice,» 2007. [En línea]. Available: http://www.sei.cmu.edu/productlines/frame_report. [Último acceso: Febrero 2010].
- [32] J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen y J.-M. DeBaud, «PuLSE: A Methodology to Develop Software Product Lines,» de *SSR '99 Proceedings of the 1999 symposium on Software reusabilit*, Los Angeles, California, USA, 1999.
- [33] ESI, «Families,» 2003. [En línea]. Available: <http://www.esi.es/Families/>. [Último acceso: Junio 2012].
- [34] F. van der Linden, «Family Evaluation Framework overview & introduction,» 2005.
- [35] M. C. Camacho Ojeda y J. A. Hurtado Alegria, «Analizando la viabilidad de adoptar el enfoque de Líneas de Productos de Software en pequeñas entidades,» de *7th Colombian Computing Congress*, 2012.

