

Experiencias de Desarrollo de Líneas de Productos Software en el Contexto de un Curso de Ingeniería de Software.

Marta Cecilia Camacho / Grupo I+D en Informática
 Profesor Asistente
 Institución Universitaria Colegio Mayor del Cauca
 cecamacho@unimayor.edu.co

Julio Ariel Hurtado Alegría / Grupo IDIS
 Profesor Asociado
 Universidad del Cauca
 ahurtado@unicauca.edu.co

Pablo H. Ruiz / Grupo I+D en Informática
 Profesor Ocasional
 Institución Universitaria Colegio Mayor del Cauca
 pruiz@unimayor.edu.co

Resumen: Las líneas de productos de Software – SPL, se han convertido en una estrategia valiosa para incrementar la competitividad de las empresas productoras de software que trabajan en dominios específicos. Sin embargo, la aplicación de dicho enfoque requiere grandes competencias técnicas, conocimientos del dominio, recursos y tiempo. Estos factores se tornan críticos cuando la entidad que desea adoptar este enfoque es pequeña y sus recursos son limitados. En este artículo se presenta un estudio de caso en el que se aplicaron y evaluaron un conjunto de prácticas recomendadas por dos de los patrones definidos en el framework propuesto por el SEI, en un proyecto de desarrollo en el contexto de una pequeña entidad dentro de un curso de Ingeniería de Software. Como principal resultado se identificó un conjunto de prácticas que por su facilidad y utilidad pueden ser incorporadas en el proceso de desarrollo SPL en el contexto de pequeñas entidades de software.

Palabras clave: Líneas de productos software, modelos de procesos, patrones de proceso, pequeñas entidades de desarrollo.

Introducción

Las empresas desarrolladoras de software se enfrentan actualmente al doble reto que exige la competitividad, mejorar la calidad de sus productos y responder a tiempos de respuesta cada vez más cortos en un mercado exigente y cambiante. Este es un gran desafío, en especial, para las empresas de software pequeñas donde los recursos y la capacidad de producción son limitados [1]. Las Líneas de Productos Software (Software Product Lines SPL) son una estrategia de producción basada en la reutilización planificada [2] [3] mediante la cual, las empresas logran disminuir costos y tiempo de salida al mercado, así como aumentar la calidad de los productos pertenecientes a la línea.

La mayoría de los casos exitosos de adopción de SPL, han correspondido a grandes empresas productoras de software [4], sin embargo, la industria del software está conformada especialmente por medianas y pequeñas empresas [1]. Aunque el enfoque es atractivo, no es fácil de adoptar, exige vencer barreras asociadas a la resistencia al cambio organizacional y en el enfoque productivo [5] [6].

Además, este enfoque requiere de inversión, tiempo y una fuerte estructura organizacional, factores críticos para las pequeñas entidades de software [6] [7]. Estos aspectos hacen que una empresa dude y decline en su intento de adoptar el enfoque SPL si no existen claros objetivos de negocio y un camino poco riesgoso a seguir. Existen reportes de algunos avances en la adopción del paradigma SPL en pequeñas organizaciones, la mayor parte de ellos se han centrado en subdominios específicos de las SPL sin alcanzar la adopción completa del paradigma [4] [8] [2] [10] [11], sin embargo, la literatura reporta algunos casos [9] [12] [13] [24] en los cuales, el enfoque ha sido implementado de forma exitosa. Estos casos que se han dado en empresas con un mercado focalizado, son una evidencia de que el enfoque es aplicable en pequeñas organizaciones, si éstas trabajan en un dominio específico. La dificultad con estos casos radica en que se reporta la aplicación del enfoque SPL, pero no se detallan los aspectos suficientes para lograr replicar estas experiencias en otros contextos.

En este artículo se explora la aplicabilidad de un conjunto de prácticas SPL en el contexto de un pequeño grupo de desarrollo, se consideró como referente el framework

para SPL propuesto por el SEI [3] y se seleccionó un conjunto mínimo de prácticas básicas. Para poder explorar las prácticas SPL aplicables en las pequeñas empresas productoras de software fue necesario primero estudiar sus características [1], para ello se revisaron reportes y estudios de la industria del software [14] [15] [16] y se realizó un diagnóstico de la industria local específico a SPL [17], a partir de los cuales se seleccionaron los patrones y las prácticas específicas que se emplearon y analizaron en éste trabajo.

Estas prácticas fueron aplicadas en la construcción de una línea de productos por una pequeña entidad de desarrollo en el marco de un curso avanzado de Ingeniería de Software en la Institución Universitaria Colegio Mayor del Cauca. La entidad presenta las limitaciones de tiempo, recursos y personal de forma similar a las limitaciones de una pequeña empresa de software. Este estudio exploratorio ha permitido discriminar un conjunto de prácticas SPL de acuerdo a su facilidad y utilidad con el fin determinar su incorporación en un proceso de desarrollo de SPL orientado a pequeñas entidades de software.

El resto de este artículo se ha organizado de la siguiente forma: la Sección II describe los trabajos relacionados a este estudio. La Sección III describe el diseño del estudio, mientras la Sección IV presenta los resultados obtenidos y su análisis respectivamente. Finalmente la sección V presenta las conclusiones, limitaciones y el trabajo futuro.

Trabajos relacionados

Las empresas de software buscan disminuir los costos de producción y acelerar el tiempo de lanzamiento de productos al mercado y precisamente son estos los beneficios esperados de la adopción del paradigma de SPL. La literatura ha planteado varios lineamientos con el fin de facilitar la adopción de este enfoque, los cuales han sido definidos en el contexto de grandes empresas capaces de asumir organizativa y económicamente las actividades de la ingeniería de dominio en el desarrollo de LPS.

Tanto el SEI (Software Engineering Institute) como el European Software Institute (ESI) y el Fraunhofer Institute for Experimental Software Engineering (IESE) han planteado iniciativas que buscan mejorar los procesos de la industria del software a través de la introducción del enfoque de líneas o familias de productos. En Europa, el ESI propone PuLSE, una metodología que permite la concepción y despliegue de las familias de productos

software [18], el SEI presenta un framework para la práctica de SPL fomentado por las experiencias de investigadores y el sector industrial [3] (3), ESI desarrolla tres proyectos (ESAPS, CAFÉ y FAMILIES), los cuales consideran desde conceptos, métodos, aplicación y evaluación de familias de software [19] [20] [21] [22].

En la literatura se encontraron algunos reportes sobre SPL en medianas y pequeñas empresas. En el marco del proyecto “Software Variant Building” del IESE [5] se aplicaron conceptos de la ingeniería de línea de productos en seis empresas pequeñas enfocado a la granularidad de las características y la variabilidad. En el trabajo de Knauber [12] [23] se presenta un caso exitoso en una pequeña empresa alemana, con experiencia en un dominio específico y con un grupo con dedicación exclusiva, plantearon las dificultades en la gestión del cambio y los requisitos, se presenta también un intento por combinar prácticas ágiles con SPL [10], se encontraron dos artículos referentes a la aplicación de SPL en empresas desarrolladoras de juegos para móviles, realizándose adaptaciones específicas [9] [24]. La mayoría de las experiencias de SPL en pymes corresponden a primeros acercamientos y no a casos de adopción o institucionalización del enfoque, los procesos propuestos han sido exitosos para grandes empresas, sin embargo, no se cuenta con un proceso disponible y orientado a las pequeñas empresas, que permita formular un conjunto de prácticas que facilite la adopción del enfoque de SPL, y así lograr incrementar los niveles de reutilización y aprovechar las ventajas que esto conlleva.

Gannod [25] narra la experiencia de incluir líneas de productos en un curso de Ingeniería de sistemas embebidos de la Universidad Estatal de Arizona, desarrollando una línea de productos de automatización del hogar, el equipo de trabajo incluyó asistentes de doctorado y maestría, docentes y estudiantes de semestres avanzados de ingeniería de sistemas y eléctrica. Los conceptos y prácticas de SPL se basaron en el framework del SEI y aplicaron el patrón “*fábrica*” [30]. Los autores consideran exitosa la experiencia por la inclusión de nuevas tecnologías en el currículo académico, además el uso del enfoque de SPL facilitó la definición de proyectos y laboratorios. En este trabajo se analizan los patrones “*Qué Construir*” y “*Construir Producto*”, los cuales están inmersos en el patrón “*fábrica*”, sin embargo el trabajo de Gannod está enfocado a la evaluación del impacto de nuevos paradigmas en el currículo de ingeniería de sistemas embebidos y no a la evaluación de la aplicabilidad de éstas prácticas en pequeñas entidades de software.

Aplicación de prácticas SPL en una Pequeña Entidad.

El objetivo de este estudio de caso fue identificar las prácticas de SPL que pueden introducirse en un pequeño grupo de desarrollo de software. Se buscó responder la pregunta ¿Qué prácticas de SPL deben considerarse para hacer operante SPL en un contexto de pequeñas entidades desarrolladoras de software? De acuerdo a la clasificación de Runeson [26] el estudio de

Tabla 1. Patrón SPL Framework ¿Qué construir?			
Prácticas del patrón	Prácticas específicas del patrón	Prácticas seguidas en el estudio de caso	Producto de trabajo
Análisis de mercado	<ul style="list-style-type: none"> Identificar fuentes de información Recolectar información Identificar segmentos de consumidores Mapear productos/segmento Examinar competencia 	<ol style="list-style-type: none"> Identificar fuentes de información Recolectar información Identificar consumidores 	<ol style="list-style-type: none"> Descripción del dominio Listado de posibles consumidores
Entendiendo dominio relevante	<ul style="list-style-type: none"> Identificar contexto dominio Identificar problemas recurrentes Capturar y representar información del dominio 	<ol style="list-style-type: none"> Identificar contexto dominio Identificar problemas Capturar y representar información dominio. 	<ol style="list-style-type: none"> Descripción del dominio. Listado de problemas Modelo de dominio
Construcción de un modelo de negocio	<ul style="list-style-type: none"> Plantear caso de negocio Estimación costos y riesgos Estimación de beneficios 	No aplica	No aplica
Previsión tecnológica	<ul style="list-style-type: none"> Direccionar grupos Fuentes de tecnología Validando pronóstico 	No aplica	No aplica
Alcance	<ul style="list-style-type: none"> Examinar productos existentes Discutir metas y productos Diagramar el contexto Hacer Matriz atributos /Productos Desarrollando escenarios Listar características /productos potenciales 	<ol style="list-style-type: none"> Examinar productos Discutir metas y productos de la línea Listar características /productos potenciales Hacer Matriz características/ Productos 	<ol style="list-style-type: none"> Listado productos existentes Descripción de la línea Alcance de la línea Matriz características /productos
		Número total: 10	total de productos: 9

Tabla 2. Patrón SPL Framework - Construir producto			
Prácticas del patrón	Prácticas específicas del patrón	Prácticas seguidas en estudio de caso	Producto de trabajo
Ingeniería de Requisitos	<ul style="list-style-type: none"> Técnicas de análisis de dominio (FODA) (33) Modelado de perspectivas Modelado de casos de uso 	<ol style="list-style-type: none"> Modelar características Modelar casos de uso 	<ol style="list-style-type: none"> Modelo de características Modelo casos de uso Diagrama de estados
Definición de arquitectura	<ul style="list-style-type: none"> Definición de la arquitectura Estilos y patrones arquitectónicos Guía construcción de productos Mecanismos variabilidad Documentar arquitectura 	<ol style="list-style-type: none"> Definición de la arquitectura Guía para construcción Mecanismos variabilidad Documentar arquitectura. 	<ol style="list-style-type: none"> Diagrama de clases (variabilidad) Guía de construcción
Evaluación de la arquitectura	<ul style="list-style-type: none"> Definición de objetivos y atributos de calidad Evaluar una arquitectura en construcción. ADR 	No aplica	No aplica
Desarrollo de componentes	<ul style="list-style-type: none"> Mecanismos de variación Desarrollo de componentes Desarrollo de nuevos componentes 	<ol style="list-style-type: none"> Mecanismos de variación Desarrollo de componentes 	<ol style="list-style-type: none"> Diagrama de clases Componentes base Componentes producto
Pruebas	<ul style="list-style-type: none"> Pruebas unitarias, Pruebas de integración Pruebas Producto Pruebas de despliegue 	<ol style="list-style-type: none"> Pruebas producto 	<ol style="list-style-type: none"> Resultados de pruebas
Integración del sistema software	<ul style="list-style-type: none"> Descripción de interfaces Ensamblaje Middleware 	<ol style="list-style-type: none"> Descripción de interfaces 	<ol style="list-style-type: none"> Diagrama de clases Núcleo base Producto
		Número total: 21	Total productos: 21

caso es exploratorio, permite extraer las prácticas desde la aplicación empírica buscando definir una hipótesis basada en la definición de un proceso SPL, este estudio de caso es de tipo holístico de acuerdo a Yin [27].

a. Selección de las Prácticas a Aplicar: El framework propuesto por el SEI abarca ingeniería de software, gestión técnica y gestión organizacional, propone un total de 29 prácticas, agrupadas en doce patrones, los patrones indican cómo agrupar estas prácticas para resolver problemas recurrentes en la adopción de acuerdo a características de las empresas [28], la revisión de estudios sobre pequeñas empresas productoras de software y el diagnóstico de la industria local [17] permitieron seleccionar los patrones enfocados a identificar el alcance para una primera línea y el desarrollo de los productos a partir de la reutilización

b. Diseño del Estudio de Caso: de acuerdo al objetivo del caso de estudio, se diseñaron los indicadores, métricas e instrumentos para identificar las prácticas de mayor valor para VSE a emplear, la tabla 3 relaciona estos elementos para el estudio de caso exploratorio.

c. Desarrollo del caso: Para la aplicación de Small SPL, los estudiantes recibieron una capacitación inicial 15 horas dividida en 2 partes. La primera parte abordó la temática de líneas de productos software, la segunda parte se enfocó en presentar el marco del SEI, sus actividades y prácticas a seguir, así como algunas técnicas y notaciones para modelar líneas de productos, la tercera y última parte correspondió al desarrollo de una primera línea de productos software. El proyecto requirió de un esfuerzo total de 98 horas por integrante para un total de 588 horas.

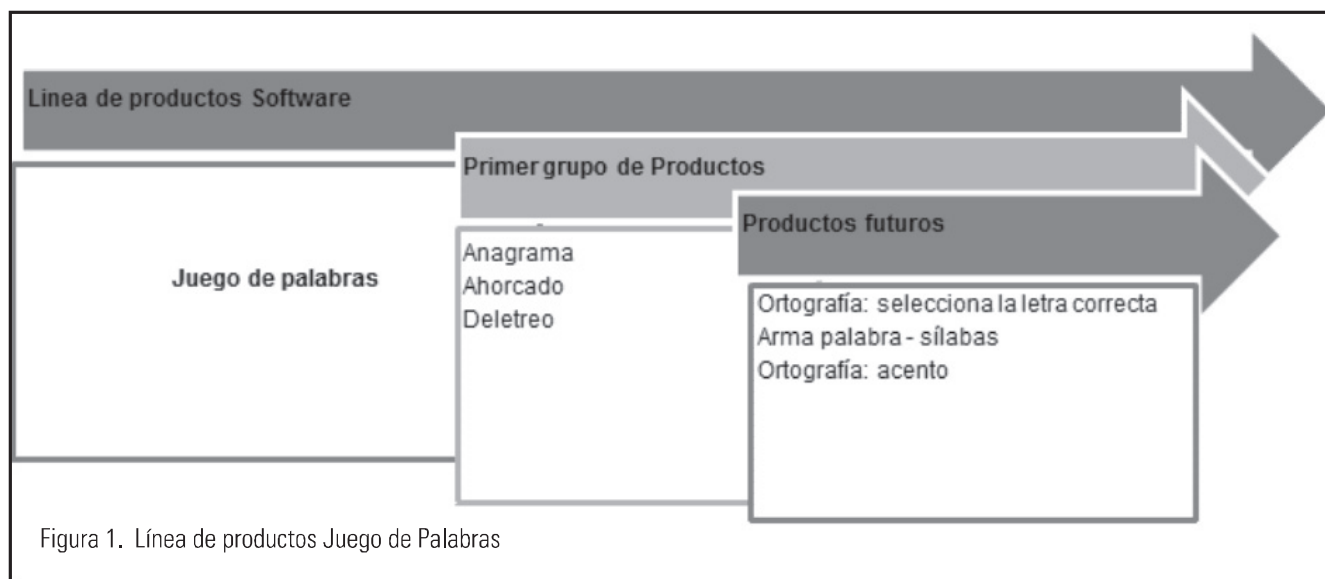
Tabla 3. Diseño del estudio de caso exploratorio

Objetivo	Indicadores	Métricas	Instrumentos
Identificar las prácticas de SPL que pueden introducirse en un pequeño grupo desarrollador de software	Nivel de dificultad de las prácticas. (Alto: NE ? 3 Bajo: NE>3)	NE - Número de ensayos para la realización correcta de un artefacto	<ul style="list-style-type: none"> Documento de alcance de la línea Modelo de características Planilla de Control de Avance
	Nivel de Utilidad de las prácticas (AU/AT%)	AU - Número de artefactos utilizados en etapas posteriores AT Número Total de Artefactos de la práctica	<ul style="list-style-type: none"> Planilla de Control de Avance

por empresas sin experiencia en el desarrollo de SPL, los patrones elegidos fueron: "Que construir" y "Construir Producto" [3] [29], de manera similar se revisaron las prácticas específicas y se seleccionaron las aplicables en este estudio de caso, en las tablas 1 y 2 se detallan patrones, practicas especificas, las practicas que se siguieron y los productos de trabajo de cada práctica.

Contexto del Estudio de Caso: El grupo de desarrollo se constituyó por seis estudiantes del curso de ingeniería de software, ofertado como electiva de último semestre en el programa tecnológico de desarrollo de software. El equipo de trabajo se caracterizó con un nivel medio de conocimientos y experiencia de desarrollo en el dominio y se consideró baja la competencia de desarrollo frente al dominio. Por la experiencia del grupo el proyecto se realizó en java y como entorno de desarrollo se empleó NetBeans.

d. Dinámica del proyecto: El desarrollo de la línea de productos en la ingeniera de dominio se especificó la línea a desarrollar, su alcance, se diseñó su arquitectura y se construyó el núcleo base, desde el cual se construyeron tres productos específicos de la línea. La ingeniería de dominio se desarrolló con todo el equipo de desarrollo, la línea identificada se denominó la línea de Juegos de palabras (se conoce como juegos de palabras a los pasatiempos que emplean palabras y/o letras para efectos lúdicos y educativos, favoreciendo el fortalecimiento de habilidades del lenguaje). Una vez establecida la línea se procedió a estudiar y analizar el dominio, con las prácticas planteadas por el patrón "que construir". El equipo siguiendo técnicas de análisis de dominio logró identificar el alcance de la línea, caracterizar los productos y seleccionar los primeros productos a desarrollar. La documentación, diseño, programación y pruebas fue dividida y asignada de



acuerdo a los conocimientos y competencias de cada uno de los integrantes. El esfuerzo requerido en la ingeniería de dominio fue de 44 horas integrante.

Para el desarrollo de los productos específicos se dividieron subgrupos de desarrollo por producto, cada uno conformado por 2 integrantes, y se aplicó el patrón “*Construir producto*”, el cual parte de los activos construidos en la ingeniería de dominio. El esfuerzo de la ingeniería de aplicaciones fue de 39 horas por integrante aproximadamente. Durante el desarrollo del proyecto se recolectó la información siguiendo el diseño del estudio por medio de una planilla de plan de avance.

e. Resultados Obtenidos: Durante la fase de la ingeniería de dominio el grupo logró trabajar en equipo, aprovechando los espacios en conjunto, esta fue una fortaleza del estudio de caso. La concepción inicial de una línea de productos fue difícil para los estudiantes acostumbrados a considerar un solo producto e inicialmente intentaban direccionar las prácticas a un solo producto, por el contexto académico empleado los estudiantes no tienden a considerar prácticas del caso de negocio o examinar productos existentes en el mercado, por lo cual fue necesario repetir algunas las prácticas del análisis de mercado.

Las prácticas consideradas para entender el dominio relevante, evidenciaron un vacío para identificar los problemas recurrentes ocasionados por falta de experiencia y conocimiento en el dominio, el grupo desarrollador empleó más el documento de descripción del dominio que el modelo de dominio. Las prácticas de identificación de dominios relevantes y del alcance fueron las de mayor dificultad dentro del patrón ¿Qué Construir?, pero también

las de mayor reutilización en prácticas posteriores. Según los comentarios de los estudiantes fue complicado pensar en varios productos al tiempo.

El caso de estudio evidenció debilidades en las prácticas de diseño y mecanismos de variación, algunos subgrupos modificaron la estructura planteada presentándose dificultades para el intercambio y uso de componentes desarrollados por otros integrantes, existió duplicación de algunos componentes. La ingeniería de requisitos se abordó combinando el modelo de características con el diagrama de casos de uso como técnica conocida, inicialmente el modelo de características (features) causó confusión, pero una vez apropiado, desplazó por completo al modelo de casos de uso, los estudiantes consideraron que el modelo de características permitió observar de manera general la línea, identificar características comunes y variables, permitiendo entender mejor la concepción de la línea como tal y que fue muy útil también para la derivación de cada producto. La tabla 4 relaciona los resultados cuantitativos obtenidos en el estudio de caso exploratorio. Estos permiten ver que las prácticas de análisis de dominio son medianamente complejas de aplicar, pero resultan significativamente útiles. Por otro lado, las prácticas del diseño del dominio resultaron más complejas de adaptar y finalmente su utilidad en la construcción de los productos resultó baja.

f. Análisis de Resultados: El estudio de caso exploratorio permitió realizar examinar el enfoque de SPL de una manera más práctica y enfocada a las pequeñas entidades desarrolladoras de software, se pudo evidenciar que una de las mayores dificultades que enfrentará una pequeña entidad es identificar una línea de productos

Tabla 4. Resultados obtenidos del caso de uso exploratorio

Prácticas	Productos	NE	Nivel de dificultad	AU/AD	Nivel de Utilidad
Análisis de mercado					
1. Identificar fuentes de información			Medio	2/2	100%
2. Recolectar información	Descripción del dominio	2	Medio		
3. Identificar consumidores	Listado de posibles consumidores	1	Bajo		
Entendiendo dominio relevante					
4. Identificar contexto dominio	Descripción del dominio	2	Medio	2/4	50%
5. Identificar problemas recurrentes	Listado de problemas	4	Alto		
6. Capturar y representar información dominio.	Glosario	2	Medio		
Alcance					
7. Examinando productos existentes	Listado productos existentes	3	Medio	3/4	75
8. Discutir metas y productos de la línea	Descripción de la línea	3	Medio		
9. Listar características/productos potenciales	Alcance: Listado de productos	5	Alto		
10. Hacer Matriz características/Productos	Matriz características/productos	4	Alto		
Ingeniería de Requerimientos					
11. Modelar de características	Modelo de características	4	Alto	2/2	100%
12. Modelar casos de uso	Modelo de casos de uso	2	Medio		
Diseño del Software					
13. Definición del Diseño	Diagrama de clases (General)	3	Medio	1/3	33%
14. Guía para la construcción	Guía de construcción	4	Alto		
15. Determinar Mecanismos de variabilidad	Modelo de Clases	4	Alto		
16. Comunicar y documentar el diseño.	Diagrama de clases	4	Alto		
Desarrollo de componentes					
17. Mecanismos de variación	Diagrama de clases	4	Alto	1/2	50%
18. Desarrollo de componentes	Componentes base	4	Alto		
Pruebas					
19. Pruebas producto	Resultados de pruebas	2	Medio	3/4	75%
Integración del sistema software					
20. Descripción de interfaces	Diagrama de clases (Interfaces)	4	Alto	1/2	50%
21. Integración del sistema	Núcleo base	4	Alto	6/10	63%

potencial y caracterizarlos, especialmente si no tiene un mercado focalizado, le exigirá tener experiencia en el dominio o dedicar una parte considerable de tiempo y recursos en esta tarea, por ello explorar que prácticas y productos deben realizar en esta etapa del desarrollo de una línea de productos fue muy importante y confirma la necesidad de un proceso de desarrollo de SPL enfocado a las pequeñas entidades de desarrollo.

El caso de estudio también permite evidenciar que las pequeñas entidades pueden tener problemas para enfrentar el diseño de software, para poder plantear adecuadamente los puntos y mecanismos de variación necesarios para una reutilización planificada. Esta parte deja clara la necesidad de detallar prácticas que acompañen la definición y diseño de los puntos y mecanismos de variación, así como la gestión de activos.

4. Conclusiones y Trabajo Futuro

Para el desarrollo de este trabajo se utilizaron varias fuentes de información. Inicialmente se estudió la literatura con el fin de entender las características de las pequeñas empresas productoras de software y se contrastaron con los hallazgos obtenidos en el estudio focalizado la industria de software en la ciudad de Popayán, entender

estas características identificar similitudes y diferencias con el equipo de desarrollo, permitiendo validar los hallazgos realizados. Además en el estudio de las pequeñas empresas productoras de software se soportó la revisión y selección de los patrones y prácticas específicas a examinar en el framework para SPL del SEI.

La valoración de prácticas para el desarrollo de una línea de productos realizado en un entorno académico logró evaluar ampliamente las prácticas relacionadas con la definición, diseño e implementación de una línea de productos, sin exigir los cambios organizaciones que debe afrontar una pequeña entidad de software ni asumir los riesgos económicos que implicaría.

El estudio de caso evidencia que es necesario un proceso que permita a las pequeñas entidades desarrolladoras de software seguir el enfoque de líneas de productos, si bien el framework plantea practicas no solo de ingeniería de software sino también de gestión, y se proponen patrones, estos no constituyen un proceso como tal, las empresas pequeñas requieren un mayor acompañamiento y una orientación más prescriptiva para sus primeros acercamientos.

Referencias

- [1] Richardson L.; Wangenheim G.; "Guest Editors' Introduction: Why are Small Software Organizations Different?," IEEE Software, vol. 24, no. 1, Feb. 2007, pp. 18-22.
- [2] Knauber P.; Succi G.; Proceedings of Software Product Lines: Economics, Architectures, and Implications, International Conference on Software Engineering (ICSE), Vol. 1.0, págs. 83-90. IESE-Report No. 070.00/E , 2000.
- [3] Northrop L.; Clements P.; et al. "A Framework for Software Product Line Practice", Version 5.0. Software Engineering Institute (SEI), Disponible en: http://www.sei.cmu.edu/productlines/frame_report/index.html
- [4] Knauber P.; Muthig D. et al. "Applying product line concepts in small and medium sized companies Software", IEEE, vol 7, no 5, (2000): 88-95
- [5] Northrop, L y Jones, L. "Introduction to software product line adoption", 13th International Software Product Line Conference (SPLC), Software Engineering Institute (SEI), 2009
- [6] Five years of product line engineering in a small company. Verlage, M y Kiesgen, T. 2005. Proceedings of the 27th international conference on software engineering.
- [7] Case Studies for Method and Tool Evaluation. Kitchenham, Barbara, Pickard, Lesley y Pfleeger, Shari. 4, 1995, IEEE Software Magazine, Vol. 12 , págs. 52-62
- [8] Ingeniería de requisitos basada en reutilización: una propuesta de aplicación a los sistemas teleoperados para limpieza de cascos de buques. Nicolás, Joaquín, y otros. Almagro : s.n., 2005. III Jornadas de trabajo DYNAMICA.
- [9] Alves, C.; Camara, T.; Alves, V. "Experiences with Mobile Games Product Line Development at Meantime", Software Product Line Conference, 2008. SPLC '08. 12th International, On page(s): 287 - 296
- [10] Hanssen G.; Fígri T.; Process fusion: An industrial case study on agile software product line engineering, Journal of Systems and Software, Vol. 81, 2008, pág. 843- 854.
- [11] Mærsk-Møller, H.; Nørregaard, B.; Experiences initiating software product line engineering in small teams with PULSE. Innsbruck, Austria : s.n., 2010. In Proceedings of IASTED International Conference on Software Engineering. Vol. SE'07. ACTA Press, 2010, págs. 125--134.
- [12] Gacek, C; et al, A Case Study: Successful Software Product Line Development in small organization. 2001. IESE Technical Report No. 013.01/E.
- [13] Sellier, D.; et al, Introducing Software Product Line Engineering for Metal Processing Lines in a Small to Medium Enterprise. s.l. : IEEE, 2007. 11th International Software Product Line Conference. págs. 54-62. ISBN:0-7695-2888-0.
- [14] Mayer R.; Panorama de la Industria. Latinoamericana de Software. Brasil, Informática LTDA. s.l. : MBI, 2004.
- [15] Fedesoft. Importancia de la industria del software a nivel mundial . Colombia, 2006.
- [16] Heshusius K.; Colombia: Desafíos de una industria en formación. [ed.] Paulo Bastos Tigre y Felipe Silveira Marques. Desafíos y oportunidades de la industria del software en América Latina. s.l. : CEPAL (Comisión Económica para América Latina y el Caribe), Mayol, 2009, Capítulo 5, págs. 139-170. <http://www.eclac.org/publicaciones/>.
- [17] Camacho, M.; Hurtado, J.; Analizando la viabilidad de adoptar el enfoque de Líneas de Productos de Software en pequeñas entidades. IEEE explorer, 2012. 7th Colombian Computing Congress.
- [18] Bayer, J. et al; PuLSE: A Methodology to Develop Software Product Lines. Los Angeles, California, USA : ACM, 1999. SSR '99 Proceedings of the 1999 symposium on Software reusabilit., págs. 122-131.
- [19] ESI. Proceso ESAPS (Engineering Software Architectures, Processes and Platforms for System-Families). [En línea] 2001. <http://www.esi.es/esaps/>.
- [20] ESI. Families. European Software Institute. [En línea] 2003. <http://www.esi.es/Families/> <http://www.esi.es/en/Projects/esaps> www.esi.es/Cafe.
- [21] van der Linden, F.; Software product families in Europe: the Esaps & Cafe projects.. 4, 2002, IEEE software, Vol. 19. 10.1109/MS.2002.1020286.
- [22] Van der Linden, F.; Family Evaluation Framework overview & introduction. ESI. s.l. : ITEA, Philips, 2005. Eureka SI 2023 Programme, ITEA project ip02009. Final – PH-0503-01– v-1.0.

- [23]Knauber, P., Gacek, C. y Schmid, K.; Successful Software product line development in a small organization. [aut. libro] Paul Clements y Linda Northrop. Software product lines, practices and patterns. Addison Wesley, 2002, págs. 485-519.
- [24]Myllärniemi, Varvana, Raatikainen, Mikko y Männistö, Tomi. Inter-organisational Approach in Rapid Software Product Family Development — A Case Study. Reuse of Off-the-Shelf Components, 9th International Conference on Software Reuse, ICSR 2006 . s.l. : Springer Berlin Heidelberg, 2006, págs. 73-86.
- [25]Gannod, G.; Doherty J.; Using a Product Line Approach to Develop Course Projects. [ed.] American Society for Engineering Education. 2003. Proceedings of the 2003 American Society Engineering Education Annual Conference & Exposition. Session 3460.
- [26] Runeson, P., Höst, M.; Guidelines for conducting and reporting case study research in software engineering. (D. Sjöberg, Ed.) Empirical Software Engineering, 2009, 131-164.
- [27]Yin R.; Case study research. Desing and methods, tercera edicion, Applied Social Reserch Methods Serie, vol 5, 2003.
- [28]Northrop, L.; SEI's Software Product Line Tenets.. 4 de Julio de 2002, Diario IEEE Software, Vol. 19, págs. 32-40. 0740-7459.
- [29]Clements, P.; Northrop, L.; Software Product Lines: Practices and Patterns. s.l. : Addison-Wesley, 2001. ISBN-10: 0201703327, ISBN-13: 9780201703320.
- [30]Greenfield J.; Short k.; Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools, OOPSLA '03 Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 2003, Páginas 16-27.