

Arquitectura para la Gestión del Ciclo de Vida de Líneas de Producto Software

David de Castro¹[0000-0002-9202-9075], Alejandro Cortiñas¹[0000-0002-2555-6342], Miguel R. Luaces¹[0000-0003-0549-2000], Oscar Pedreira¹[0000-0001-6176-4475], and Ángeles S. Places¹[0000-0001-8539-304X]

Universidade da Coruña, CITIC, Database Lab
{david.decastro, alejandro.cortinas, miguel.luaces, oscar.pedreira, angeles.saavedra.places}@udc.gal
<https://lbd.udc.es>

Resumen La Gestión del Ciclo de Vida de Aplicaciones (conocido como ALM por su nombre en inglés, *Application Lifecycle Management*) coordina las actividades para desarrollar, mantener y evolucionar una aplicación, definiendo un marco que busca la eficiencia y efectividad en cada etapa del ciclo de vida. Aunque existen numerosas publicaciones sobre ALM, la aplicación de ALM en las Líneas de Producto Software (LPS) aún no ha sido abordada. En este trabajo se propone adaptar ALM para el desarrollo de LPS mediante la creación de una herramienta que maneje principalmente las áreas de desarrollo y operaciones para gestionar varias LPS y sus productos asociados.

Keywords: Línea de Producto Software · Application Lifecycle Management · Feature Model

1. Introducción

Durante los últimos años, en el Laboratorio de Bases de Datos hemos trabajado en una Línea de Producto de Software (LPS) enfocada al desarrollo de Sistemas de Información Geográfica y en una segunda LPS enfocada a Bibliotecas Digitales. La gestión de ambas LPS nos plantea varios problemas sin resolver. Uno es el mantenimiento de los productos generados por la LPS ya que cada vez que se realiza un cambio en la especificación del producto es necesario generar el código y realizar una fusión del nuevo código con el antiguo mediante herramientas del repositorio de código fuente. Todo esto se hace manualmente, aunque a menudo podría ser automatizado ya que la integración del código es inmediata y no hay conflictos. Lo mismo sucede con cualquier cambio realizado en las plantillas de código de la LPS. Cuando se modifican, es necesario modificar manualmente todos los productos que se generaron previamente con esa LPS.

La solución a estos problemas es también el objetivo del paradigma de Gestión del Ciclo de Vida de Aplicaciones (conocido como ALM por su nombre en inglés, *Application Lifecycle Management*). ALM integra y gestiona diferentes



actividades, procesos, herramientas y personas involucradas en la *gobernanza, desarrollo y operaciones* en todo el ciclo de vida de productos de software, desde su concepción hasta su retiro [2]. Aunque existen aplicaciones que dan soporte a ALM, no existe actualmente ninguna que se centre en la gestión de Líneas de Producto Software y su ciclo de vida. Por ello, este artículo propone una herramienta que permite gestionar actividades de ALM para LPS.

2. Application Lifecycle Management en LPS

En la Figura 1, proponemos la arquitectura de una herramienta (SPLALM: Software Product Line Application Lifecycle Management) diseñada para crear y gestionar LPS, y para definir y generar productos a partir de ellas. Los recuadros de fondo oscuro indican las actividades ALM soportadas por cada parte de la herramienta. En esta primera versión de SPLALM no hemos abordado la implementación de todas las actividades ALM para LPS porque sería un alcance demasiado ambicioso. La arquitectura incluye dos subsistemas principales: la línea de productos de software (parte superior derecha de la Figura 1), que corresponde al ciclo de desarrollo de la línea de productos de software de ALM, y el producto (parte inferior izquierda de la Figura 1), que corresponde al ciclo de desarrollo del producto de ALM. La arquitectura también incluye componentes para la gestión de la cartera del área de gobierno de ALM (parte superior izquierda de la Figura 1). Por último, también incluimos un componente para el despliegue de los productos (parte inferior derecha de la Figura 1) que gestiona de hecho una actividad del área de operaciones de ALM.

Una de las principales funcionalidades de SPLALM es la definición e implementación de Líneas de Producto Software, que consta de dos componentes: *SPL generator* y *Code template generator*. El componente *SPL generator* da soporte a la actividad *Domain Design* de ALM para LPS y se centra en la definición de los modelos utilizados para definir los productos. Su módulo *DSL definer* permite la definición de un DSL que permite la definición de elementos de dominio específicos como mapas para Sistemas de Información Geográfica. Su módulo *FM definer* permite la creación de un modelo de características que permite la definición de funcionalidades para la familia de productos. El segundo componente, *Code template generator*, da soporte a las actividades *Domain Realization* y *Domain Testing* de ALM para SPL. Funciona como un editor de código para crear y editar el código fuente del producto generado. El conjunto de artefactos resultantes de estos dos componentes forman la LPS, que se almacena en un repositorio de código fuente (en nuestro caso, Gitlab), haciéndolo accesible a cualquiera que necesite acceder a los modelos y al código fuente desde sus equipos.

Además de definir LPS, SPLALM también proporciona soporte para generar productos y facilitar su mantenimiento a lo largo de su ciclo de vida. Los distintos componentes para trabajar a nivel de producto se encuentran en la parte inferior de la Figura 1. El componente *Product spec generator* da soporte a la actividad *Application Design* de ALM para LPS. El otro es el *Product genera-*

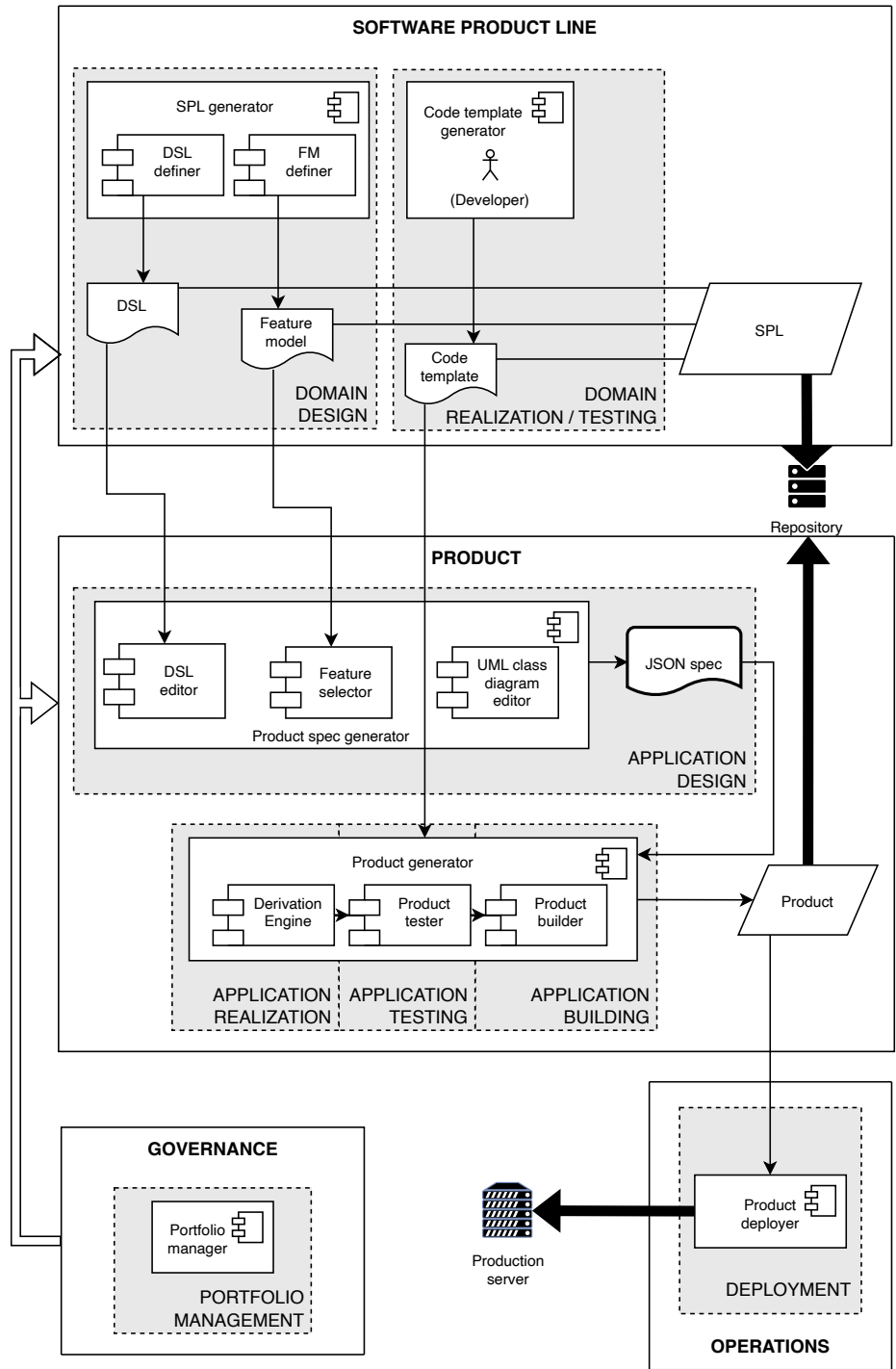


Figura 1. Arquitectura de SPLALM

tor, que soporta las actividades *Application Realization*, *Application Testing*, y *Application Building* de ALM para LPS.

Tras definir el producto, el siguiente paso es generarlo. El componente *Product generator* se encarga de esta tarea utilizando un motor de derivación [1]. Este motor genera el producto a partir de dos entradas: la especificación del producto en formato JSON y el código fuente creado por el componente *Code template generator*. El producto generado es probado por el componente *Product tester* para comprobar que funciona correctamente, luego se construye para desplegarlo, y finalmente se almacena en un repositorio remoto para permitir un fácil acceso y modificación por parte de los desarrolladores. Al mismo tiempo, es desplegado en el servidor de producción por el componente *Product deployer*, que se encuentra en el área de Operaciones.

3. Conclusiones

En este artículo proponemos la creación de una herramienta, SPLALM, que utiliza el modelo de *Application Lifecycle Management (ALM)* aplicado a Líneas de Producto Software (LPS) para administrar múltiples LPS simultáneamente, así como administrar los productos generados por ellas, de forma que algunos procesos (como la actualización o el despliegue de productos) se realicen automáticamente cuando se detecten cambios. Al describir esta herramienta, se presenta su arquitectura, los diferentes componentes que debería tener y las actividades a las que da soporte cada uno de estos componentes.

Como trabajo futuro se pretende realizar un análisis más profundo y formal de ALM para LPS, ya que el objetivo principal de este artículo era poder definir una primera versión superficial de la herramienta propuesta. El desarrollo de la herramienta SPLALM y la mejora de su definición y arquitectura para integrar todas las actividades del modelo de ALM para LPS, incluyendo actividades de gobierno, son otros de los puntos a realizar en el futuro.

Agradecimientos: TED2021-129245B-C21 (PLAGEMIS): partially funded by MCIN/AEI/10.13039/501100011033 and “NextGenerationEU”/PRTR; MAGIST: PID2019-105221RB-C41, partially funded by MCIN/ AEI/10.13039/501100011033; GRC: ED431C 2021/53, partially funded by GAIN/Xunta de Galicia.

Referencias

1. Cortiñas, A., Luaces, M.R., Pedreira, O.: Spl-Js-Engine: A JavaScript Tool to Implement Software Product Lines. In: Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume B. p. 66–69. SPLC '22, Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3503229.3547035>
2. Tüzün, E., Tekinerdogan, B., Macit, Y., İnce, K.: Adopting integrated application lifecycle management within a large-scale software company: An action research approach. *Journal of Systems and Software* **149**, 63–82 (2019), <https://doi.org/10.1016/j.jss.2018.11.021>

