



Uso de Líneas de Productos de Software en el proceso productivo de la UCI

Use of Software Product Lines in the productive process of the UCI

MSc. Lianet Salazar Labrada

MSc. Diannet Sospedra López

MSc. Reynaldo Álvarez Luna

DrC. José Felipe Ramírez Pérez

Universidad de las Ciencias Informáticas. La Habana. Cuba

Resumen

La mejora de procesos es una estrategia que se utiliza para aumentar la calidad de los productos y servicios que se desarrollan en una organización. Los modelos, estándares y normas incluyen un conjunto de actividades que permiten seleccionar, diseñar e implementar soluciones informáticas. Además, promueven la reutilización de componentes, gestión del conocimiento y la utilización de las líneas de productos durante el proceso de desarrollo de software. Durante este proceso se pueden reutilizar partes de algún software desarrollado anteriormente, y en este caso la reutilización es indispensable para el desarrollo rápido de productos y/o componentes de producto, además permite reducir los costos y riesgos asociados al desarrollo. Además, se pueden utilizar líneas de productos de software (LPS), las cuales requieren almacenar sus activos de software en repositorios para asegurar la disponibilidad de activos y apoyar el desarrollo de los productos de la línea. En la Universidad de las Ciencias Informáticas (UCI) se sigue una estructura que está dividida en LPS y la documentación que se genera en el área asociada al desarrollo de software se puede realizar en cualquiera de las estructuras definidas. El objetivo de este trabajo es mostrar los elementos que se tienen en cuenta para mantener y actualizar la política de reutilización de la



universidad, las herramientas de apoyo al proceso productivo y todos estos aspectos relacionados con la estructura de LPS que se utiliza en la UCI.

Palabras clave: líneas de productos de software, reutilización de componentes, repositorios de software; arquitectura en la LPS.

Abstract

Process improvement is a strategy that is used to increase the quality of products and services that are developed in an organization. The models, standards and norms include a set of activities that allow to select, design and implement computer solutions. In addition, they promote the reuse of components, knowledge management and the use of product lines during the software development process. During this process, parts of previously developed software can be reused, and in this case reuse is essential for the rapid development of products and / or product components, as well as reducing the costs and risks associated with development. In addition, you can use software product lines (LPS), which require you to store your software assets in repositories to ensure the availability of assets and support the development of the line's products. At the University of Informatics Science (UCI) a structure that is divided into LPS is followed and the documentation generated in the area associated with software development can be performed in any of the defined structures. The objective of this work is to show the elements that are taken into account to maintain and update the reuse policy of the university, the tools to support the productive process and all these aspects related to the LPS structure used in the UCI.

Keywords: software product lines, reuse of components, software repositories; architecture in the LPS.

Introducción

La Universidad de las Ciencias Informáticas (UCI) es un centro docente - productor que desarrolla aplicaciones y servicios informáticos orientados a diversos sectores de la economía y los servicios, dentro y fuera de Cuba, a partir de la vinculación formación - producción - investigación como modelo de formación (Mustelier, 2014). La actividad de desarrollo - producción es soportada por una red de centros, integrada por 15 centros de desarrollo. Con el propósito de asegurar la calidad de los productos y ganar en competitividad, la UCI decidió adoptar procesos para el desarrollo de software a nivel de la organización. Después de la revisión de varios estándares, normas y modelos se decidió utilizar CMMI-DEV para el desarrollo, en su versión 1.2 y en su representación escalonada el nivel 2 (Ramos et al., 2011; Ramos, 2013).

En julio de 2011, la universidad obtuvo la evaluación del nivel 2 de CMMI en los centros CEIGE (Centro de Gobierno Electrónico), CEDIN (Centro de Informática Industrial) y CESIM (Centro de Informática Médica), reconocido por el SEI de la Carnegie Mellon University; avalando la calidad del proceso de desarrollo de software que se realizaba. Luego de obtener la evaluación se decidió en el año 2015 evaluar toda la actividad productiva de la UCI con el nivel 2 de CMMI-DEV v1.3 (Concepción, 2015). A partir de entonces se trabaja para evaluar el nivel 3 donde los procesos deben estar bien definidos y descritos en estándares, procedimientos, herramientas y métodos.

Desde el 2012 la universidad ha orientado el desarrollo de software a cinco áreas temáticas: Salud, Educación, Telemática, Empresa e Industria y Administración Pública. Cada uno de los centros de desarrollo



se especializa en una de estas 5 áreas y puede agrupar sus proyectos en dominios y/o líneas de productos, lo que permite reutilizar la documentación y la arquitectura implementada para otros desarrollos. En la UCI se creó un Grupo Técnico (GT), el cual tiene como objetivo fundamental organizar y alinear todos los temas relacionados con la arquitectura en la universidad. El GT definió que el modelo de desarrollo tecnológico de la producción estará basado en la reutilización de activos de software y las arquitecturas de referencia de soluciones informáticas. Junto al GT trabaja el EPG (Grupo de Ingeniería de Procesos) encargado de definir, diseñar y documentar los procesos de desarrollo de software de la Universidad.

En la universidad se integraron tres de las áreas del nivel 3 de CMMI con el objetivo de definir un proceso de Ingeniería que tiene como propósito desarrollar la solución, a partir de la obtención de los requisitos hasta la integración y entrega del producto. Una de las áreas de proceso que integran este grupo es Solución Técnica, la cual se encarga de seleccionar, diseñar e implementar las soluciones para los requisitos. Para realizar la documentación y el desarrollo de los componentes y/o productos se han definido 2 niveles: a nivel de LPS o a nivel de Proyecto / Producto; según el objetivo de los proyectos que inicien deben utilizar uno de estos 2 niveles. Para almacenar la información de los procesos y artefactos generados por los proyectos se utiliza un expediente de proyecto que se encuentra visible en los documentos públicos del sistema de gestión de documentos administrativos XABAL EXCRIBA.

Como se ha explicado anteriormente en la UCI se sigue una estructura que está dividida en LPS y la documentación que se genera en el área asociada al desarrollo de software se puede realizar en cualquiera de las estructuras definidas. El problema de esta investigación se enfoca en cómo contribuir a la reutilización de activos de procesos y componentes, recolección de lecciones aprendidas y validación de las arquitecturas de referencia, que impacte de manera positiva en la disminución del tiempo, recursos y esfuerzo dedicados al desarrollo de los componentes y productos. Para dar solución al problema planteado se traza como objetivo de este trabajo mostrar los elementos que se tienen en cuenta para mantener y actualizar la política de reutilización de la universidad, las herramientas de apoyo al proceso productivo y todos los aspectos relacionados con la estructura de LPS que se utiliza en la UCI.

Materiales y métodos o Metodología computacional

Se utilizaron para el desarrollo de la investigación los siguientes métodos científicos, además se analizaron un conjunto de conceptos asociados al problema planteado en trabajo:

Métodos teóricos:

- **Analítico - sintético** para extraer los elementos más importantes relacionados con el objeto de estudio de la investigación y para realizar un análisis de las diferentes variantes y componentes del proceso.
- **Sistémico** para organizar y estudiar toda la información encontrada sobre el proceso Solución Técnica en los modelos, normas y estándares existentes.
- **Modelación** para realizar el diseño y modelación de los elementos y relaciones del proceso propuesto.



Métodos empíricos:

- **Entrevista** para obtener información sobre la situación problemática y la validación de los resultados.
- **Observación participante** para conocer la realidad que se estudia mediante la percepción directa.
- **Análisis comparativo:** para analizar a partir de criterios de evaluación, los elementos que tienen en común los modelos, normas y estándares revisados, además, identificar buenas prácticas en el proceso de desarrollo de los componentes y/o productos.
-

Análisis de los principales conceptos asociados al trabajo

Línea de Productos de Software

Las LPS se refieren a técnicas de ingeniería para crear un portafolio de sistemas de software similares, a partir de un conjunto compartido de activos de software, usando un medio común de producción. Las LPS son un conjunto de sistemas de software que comparten un conjunto común y gestionado de aspectos que satisfacen las necesidades específicas de un segmento de mercado o misión y que son desarrollados a partir de un conjunto común de activos de software de una manera prescrita (Espinoza & Salavert, 2016).

La LPS consiste de una familia de sistemas de software que tienen una funcionalidad común y alguna funcionalidad variable. La funcionalidad común descansa en el uso recurrente de un conjunto común de activos reutilizables (requisitos, diseños, componentes, servicios web), donde los activos son reutilizados por todos los miembros de la LPS (Miranda, Casas & Marcos, 2015). El SEI, plantea que una LPS se refiere a un conjunto de software que comparten características y que son desarrollados a partir de un conjunto de componentes reutilizables. Un enfoque de reutilización sistemático dentro de una organización aporta beneficios importantes, pero requiere de un esfuerzo significativo (SEI, 2017a).

El objetivo principal de una LPS es reducir el tiempo, esfuerzo, costo y complejidad de crear y mantener los productos de la línea (SEI, 2017b). Las LPS requieren almacenar sus activos de software en repositorios para asegurar la disponibilidad de activos y apoyar el desarrollo de los productos de la línea. La autora considera que las definiciones planteadas por los diferentes autores tienen como objetivo común que las LPS, son sistemas de software similares y que el uso de las mismas aumenta la productividad, reduciendo los indicadores asociados al tiempo, recursos y esfuerzo necesario para desarrollar un conjunto de productos de software asociados a una línea.

Arquitecturas de referencia para las LPS

Un componente clave es el uso de una arquitectura de línea de productos que permita a la organización identificar y reusar los artefactos. La arquitectura es, en ese sentido, la que asegura la cohesión de la línea de productos (Zubrow & Chastek, 2003). La arquitectura da una forma estable a todos los productos, indica los elementos de diseño que son comunes y también cómo se soporta la variabilidad de cada producto. De cierta manera, la arquitectura evoluciona a la par de los activos del dominio, e incluye componentes que sean necesarios para garantizar la variabilidad y otros atributos de calidad deseados (Vázquez, 2011).



La arquitectura de software de una LPS es un elemento fundamental en los componentes reutilizables pues es compartida por los distintos productos de una línea. La arquitectura es uno de los activos claves en cualquier LPS (Bass et. Al, 1997; Bass et. al, 2000). Esta brinda, entre otras cosas, una forma unificada de representación de los componentes y las maneras de integrarlos. La arquitectura da una estructura única a aquello que es común a todos los productos y cómo pueden variarse (Stark, 2015). Por otra parte, la arquitectura brinda a los sistemas de atributos críticos como fiabilidad, eficiencia, modificabilidad (Garlan & Shaw, 1994).

La arquitectura genérica de una LPS describe la estructura de toda la familia de productos de esa línea y captura los aspectos comunes y variables. Los aspectos comunes de la arquitectura son capturados por los componentes de software que son comunes a toda la familia. La arquitectura de dominio tiene componentes comunes a todos los miembros de la familia, componentes opcionales que son requeridos por algunos miembros y componentes variantes de los cuales algunos miembros de la familia emplean distintas versiones. El uso de las LPS en una empresa de software es un proceso complejo, sin embargo, varios modelos de procesos proponen el desarrollo de software basado en línea de productos. Uno de los modelos que menciona el uso de LPS es CMMI-DEV.

Reutilización de activos, componentes y productos de software

Los activos son los elementos que se reutilizan a lo largo de la vida del software y cada activo es desarrollado de forma particular. El Instituto de Ingeniería Eléctrica y Electrónica (IEEE) (IEEE, 2010) considera que la reutilización es el uso de un activo para la solución de diferentes problemas. Un activo reutilizable es un elemento tal como diseños, especificaciones, código fuente, documentación, unidades de prueba y manuales de procedimientos, los cuales fueron diseñados para su uso en múltiples contextos. Un activo puede ser usado en más de un software o puede ser incorporado dentro de otro active. (Vázquez, 2011)

Los activos se pueden reutilizar en diferentes etapas del ciclo de vida de un producto y permiten disminuir el esfuerzo, costo y tiempo dedicado al desarrollo. En muchas ocasiones los desarrolladores reutilizan diferentes componentes y/o activos, lo que permite incrementar la productividad y la calidad de los productos. Lo reutilización de activos y componentes, permite a los desarrolladores agilizar el desarrollo del producto y la generación de conocimiento, incorporando resultados positivos de experiencias anteriores. De esta manera se capitalizan las ventajas de la reutilización, agilizando el proceso e incrementando los niveles de calidad del producto (Barón, 2012).

Los autores de este trabajo consideran que las ideas y definiciones planteadas anteriormente coinciden en que la principal diferencia radica en el objeto de reutilización, que puede ser un componente, producto o activo. También se puede concluir que reutilizar activos, componentes o productos, es una forma de compartir y generar conocimiento. Una de las ventajas de la reutilización es que reduce la cantidad de documentación que se debe generar, pues sólo se documentan los nuevos componentes desarrollados. La reutilización, no es solo volver a utilizar funcionalidades desarrolladas y empaquetadas en clases, componentes o servicios. También puede reutilizarse el conocimiento y la experiencia de otros desarrolladores. Existen problemas que se presentan una y otra vez cuando se desarrolla un producto de software. Todo el conocimiento producido durante el desarrollo de software puede reutilizarse, recopilarse y publicarse en catálogos y repositorios.



Resultados y discusión

Comparación de los modelos, estándares y norma revisados

Para el desarrollo de la investigación, se realizó un análisis bibliográfico sobre algunos de los modelos, estándares y normas internacionales que incluyen actividades asociadas al desarrollo de los componentes y/o productos: como son MPS.BR, Competisoft, CMMI, ISO/IEC 12207 y IEEE 1074-2006. La evaluación se realiza teniendo en cuenta si los modelos, estándares o normas incluyen o proponen:

- políticas, actividades, tareas, productos de trabajo, roles y herramientas.
- actividades y/o tareas para reutilizar activos de procesos y componentes de producto.
- actividades y/o tareas para documentar las lecciones aprendidas en función de potenciar la gestión del conocimiento a nivel organizacional.
- técnicas, estándares y métodos para el diseño, implementación y documentación del componente y/o producto.
- actividades y/o tareas para las organizaciones que utilicen líneas de productos durante el desarrollo de los componentes y servicios.

Como se muestra en la tabla 1 algunos modelos y estándares coinciden en que se deben tener en cuenta actividades para reutilizar activos y componentes, almacenar lecciones aprendidas y utilizar LPS durante el proceso de desarrollo de la solución. Después del análisis realizado se identificaron las siguientes buenas prácticas:

- utilizar LPS durante la ejecución del proceso de desarrollo de software y la documentación del componente o producto.
- reutilizar activos de proceso y componentes de productos.
- recoger lecciones aprendidas relativas al proceso.

Tabla 1. Comparación entre modelos, estándares y normas revisados. Fuente: elaboración propia.

Criterios	MPS. BR	Competi- soft	CMMI	ISO/IEC 12207	IEEE 1074- 2006
Políticas	Sí	Sí	Sí	No	No
Actividades y/o tareas	No	Sí	Sí	Sí	Sí
Productos de Trabajo	No	Sí	Sí	No	No
Roles	No	Sí	No	Sí	No
Herramientas	No	No	No	No	No

Reutilización	Sí	Sí	Sí	Sí	Sí
Gestión de Conocimiento	Sí	Sí	Sí	Sí	Sí
Técnicas, estándares y métodos para el diseño e implementación	Sí	No	Sí	No	No
Uso de líneas de productos	Sí	Sí	Sí	Sí	No
Estándares de documentación	Sí	No	Sí	No	No

Finalmente se decide utilizar como modelo guía, las buenas prácticas definidas en el modelo CMMI para orientar el desarrollo del proceso Solución Técnica. Este modelo define entradas, salidas, actividades, indicadores, productos de trabajo, técnicas, estándares y métodos para el diseño, implementación y pruebas unitarias. CMMI incluye una política genérica para recoger las experiencias relativas al proceso y dar soporte a la mejora de los procesos y los activos de proceso de la organización. Además, incluye actividades específicas a realizar en caso de que la organización utilice un esquema de LPS y promueve la reutilización de activos de procesos y componentes de producto.

Documentación a realizar para la LPS

Los 14 centros de desarrollo que existen en la UCI siguen una estructura que está dividida en LPS. La documentación que se genera en el área asociada a la arquitectura del software se puede realizar en cualquiera de las estructuras definidas en la universidad y que se muestran en la figura 1. A nivel de LPS se definen los documentos: *Guía Base y Configuración*, *Componentes de Producto e Interfaces*, *Arquitectura Vista de Datos*, *Arquitectura Vista Entorno de Desarrollo Tecnológico* y *Arquitectura Vista de Presentación*. Los documentos: *Definición de Arquitectura de Software*, *Implementación del Diseño*, *Arquitectura Vista de Proceso*, *Arquitectura Vista de Sistema* se definen a nivel de producto. Es válido aclarar que los documentos que se generan a nivel de LPS se pueden actualizar a nivel de producto. En las vistas a nivel de LPS, se deben plasmar los aspectos que sean comunes para todos los productos incluidos en esa estructura. En las vistas a nivel de producto se puede reutilizar la información que existe en las estructuras superiores y sólo se debe actualizar las que sean necesarias, con la información específica de ese producto. A continuación, se muestran los elementos que se tienen en cuenta para mantener y actualizar la política de reutilización de la universidad, las herramientas de apoyo al proceso productivo y todos estos aspectos relacionados con la estructura de LPS que se utiliza en la UCI.

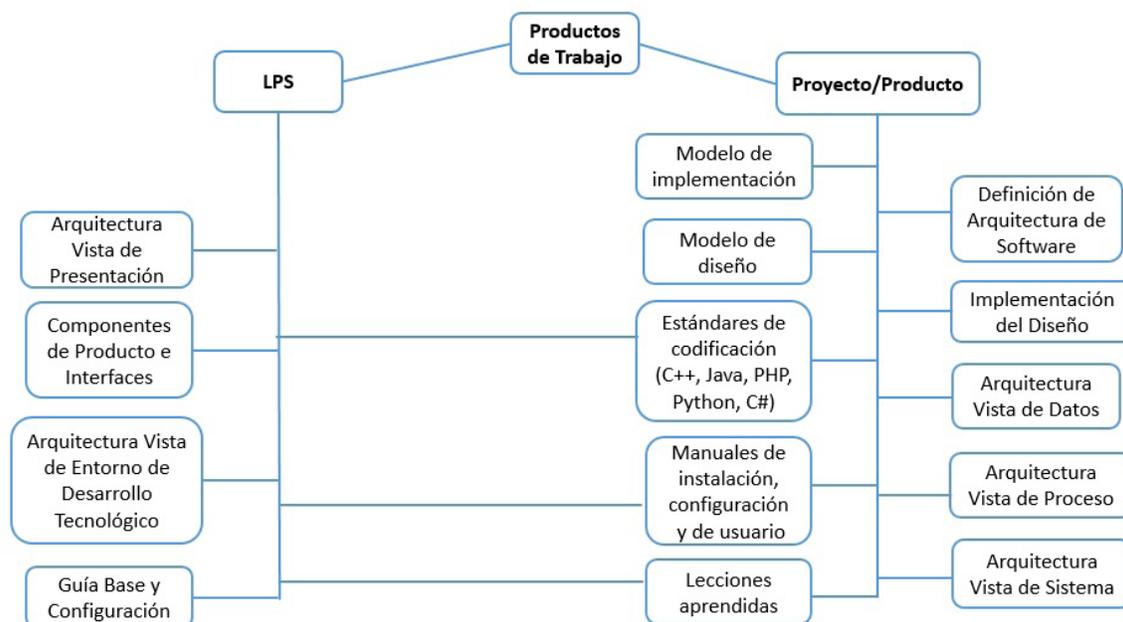


Figura 1: Productos de trabajo del área Solución Técnica. Fuente: elaboración propia.

Herramientas que brindan soporte al proceso de desarrollo de software

Para dar soporte al proceso de desarrollo de software, la gestión de los repositorios y activos de software reutilizables es necesario el uso de las siguientes herramientas establecidas en la universidad, las cuales se muestran en la figura 2:

- XEDRO GESPRO: sistema para gestionar los proyectos que se desarrollan en la universidad durante todo su ciclo de vida.
- XABAL EXCRIBA: sistema para gestionar la documentación e información que se genera en cada proyecto.
- Eclipse Process Framework Composer (EPF Composer): se utiliza para definir, documentar y publicar los procesos. Esta información se encuentra disponible en todo momento para los responsables de mantener e implementar los procesos, a través del sitio Mejora de Procesos de Software de la Universidad de las Ciencias Informáticas.
- GitLab: es donde se almacena el código de cada uno de los productos y/o componentes que se desarrollan y se mantiene a través de un control de versiones.
- Herramientas para el modelado: se utilizan para realizar el modelado de los procesos asociados al componente y/o producto que se va a desarrollar, se modelan los diferentes diagramas (casos de uso del negocio y del sistema, clases, despliegue, diseño, matrices, casos de prueba, entidad-relación), haciendo uso del Visual Paradigm.
- XEDRO GESPRO de la Dirección General de Producción (DGP): es donde se organiza la producción, estandarizando políticas de gestión de las oportunidades de proyectos, de la ejecución de proyectos y la infraestructura de desarrollo; implantando plataformas de servicios tecnológicos y el

modelo de desarrollo tecnológico de la producción basada en la reutilización de activos de software y las arquitecturas de referencia de soluciones informáticas.

- GitLab de las comunidades (codecomunidades): es público para toda la comunidad universitaria, pero es gestionado por la DGP.
- Gestores de dependencias y administradores de artefactos (Nexus y Toran): se utilizan con el objetivo de gestionar las dependencias necesarias para los diferentes proyectos y brindan la posibilidad de crear repositorios propios. En el caso de Nexus se utiliza para gestionar las dependencias de Java, .Net y Python; en el caso de Toran se utiliza para PHP.



Figura 2: Relación de las herramientas utilizadas con el proceso. Fuente: elaboración propia.

Procedimiento para la gestión de activos de software en los centros de desarrollo de la UCI.

El proceso de gestión de activos de software consta de dos niveles: Alta Gerencia y Entidad Desarrolladora (centro) y se realiza a través del uso de la Wiki del GESPRO de la DGP. A continuación, se explica cómo funciona el repositorio y catálogo de activos de software de la UCI y en la figura 3 se muestra la estructura y composición que se sigue en la UCI:

- Los activos son componente o productos ya desarrollados en los proyectos de desarrollo de la UCI.
- Todos los activos generados en el centro constituyen el catálogo de activos del centro.
- A nivel gerencial se evalúan los activos y se define si son o no reutilizables para su posterior publicación a todos los centros.
- Todos los activos son evaluados antes de registrarlos en el repositorio de activos, además se revisa

la caducidad del soporte y se actualizan o se desactivan los activos.

- En el dominio del centro de desarrollo se deben identificar los activos necesarios para los desarrollos inmediatos o futuros.

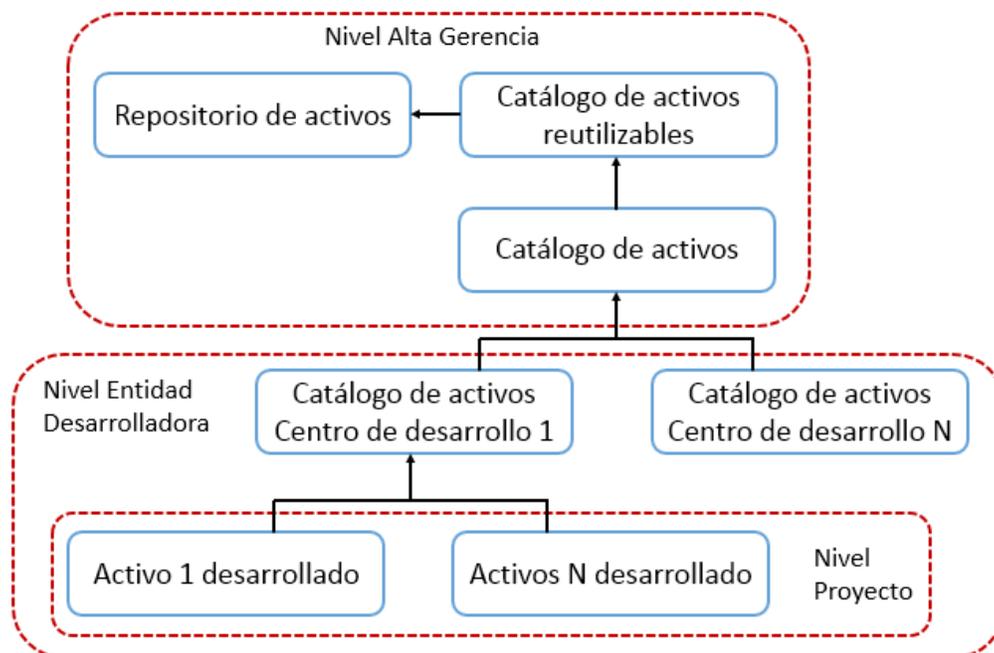


Figura 3: Procedimiento para la gestión de activos de software en los centros de desarrollo de la UCI.

En la figura 4 se muestra los estados por los que transita un activo de software reutilizable en la UCI:

- Definido: estado que adquiere una vez que ha sido publicado.
- Elaborado: estado que adquiere una vez que ha sido elaborado y terminado.
- Certificado: estado que adquiere una vez que ha sido liberado por el Laboratorio de pruebas de la Dirección de Calidad de la UCI.
- Listo: estado que adquiere una vez que ha sido elaborado, revisado por los administradores del repositorio, definido por el centro de desarrollo y certificado por el Laboratorio de pruebas de la Dirección de Calidad de la UCI.
- Reutilizable: estado que adquiere una vez que tenga todos los metadatos, la documentación requerida, haya sido revisado por los administradores del repositorio, liberado por la Dirección de Calidad y se encuentre listo para su reutilización.
- Publicado: estado inicial que adquiere una vez que ha sido subido al repositorio.
- Desactivado: estado que adquiere una vez que ha sido reemplazado y no tendrá soporte por lo que se decide desactivarlo.

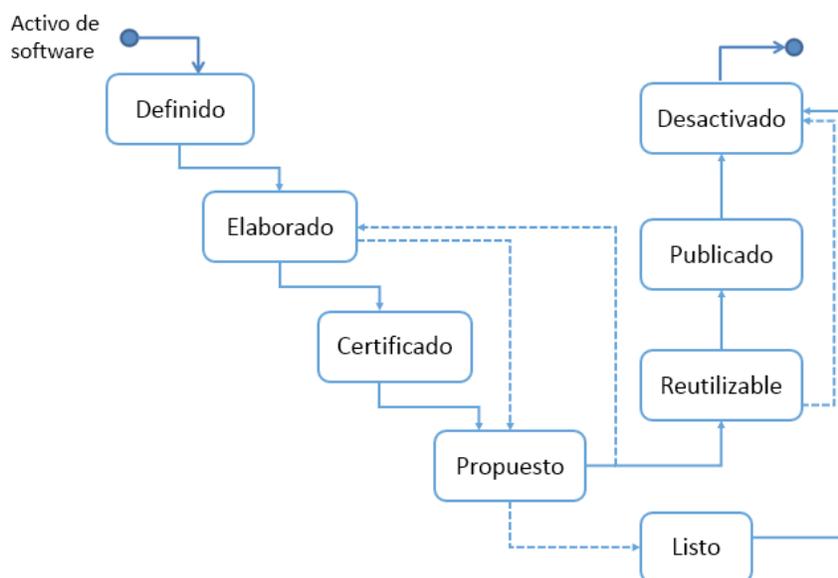


Figura 4: Estados por los que transita un activo de software reutilizable en la UCI.

Requisitos que debe tener un activo de software reutilizable

Se establece además que un activo de software reutilizable debe tener los siguientes requisitos para que pueda ser reutilizado:

- **Identificable:** debe tener una identificación que permita acceder fácilmente a sus servicios, su clasificación y búsqueda en repositorios de componentes.
- **Independiente o Auto contenido:** debe depender lo menos posible de otros componentes para cumplir su función de forma tal que pueda ser desarrollado, probado, optimizado, utilizado, entendido y modificado individualmente.
- **Reemplazable por otro componente:** se pueda reemplazar por nuevas versiones u otro componente que lo reemplace y mejore.
- **Accesible solamente a través de su interfaz:** debe exponer al público únicamente el conjunto de operaciones que lo caracteriza (interfaz) y ocultar los detalles de su implementación.
- **Bien documentado:** debe estar correctamente documentado, incluyendo sus servicios, para facilitar su búsqueda si se quiere actualizar, integrar con otros o adaptarlo.
- **Independiente de la plataforma:** debe poder ejecutarse en diversos sistemas operativos.

Finalmente se establece como política de reutilización para toda la universidad que el repositorio de activos incluye todos los activos publicados en cada una de las instancias del GESPRO de los centros de desarrollo a través de la sincronización de todos en el GESPRO Gerencial. De cada activo se publica solo los metadatos para garantizar la búsqueda, el repositorio donde se encuentran los artefactos se encuentra protegido o público de acuerdo a los intereses de la producción y del centro que genera el activo y está almacenado en alguno de los sistemas para el control de versiones disponibles, o desde un gestor de artefactos de los mencionados anteriormente.

Conclusiones

Con el trabajo realizado se puede concluir que el uso de las LPS aumenta la productividad, reduciendo los indicadores asociados al esfuerzo, coste y tiempo necesario para desarrollar un conjunto de productos de software asociados a la línea. Esta práctica promueve la reutilización de activos, componentes y/o productos de software, además del uso de repositorios y catálogos de activos que sirvan como base para gestionar el conocimiento en una organización que utilice algún modelo, norma y/o estándar de calidad para mejorar sus procesos. Además, se mostraron los productos de trabajo y las herramientas de apoyo al proceso productivo de la universidad, lo cuales en su conjunto forman parte del proceso de desarrollo de las soluciones técnicas. Se describió el procedimiento que se realiza en la universidad para la gestión de activos de software en cada uno de los centros de desarrollo, además se mostraron los requisitos que debe tener un componente para que sea reutilizado y los estados por los que debe transitar un activo durante su ciclo de vida útil. Todos estos elementos forman parte de la política de reutilización que se debe utilizar y que está definida en la universidad, con el objetivo de realizar una correcta gestión de los componentes de software reutilizables desarrollados en cada uno de los centros.

Referencias

- Barón Salazar, A. Á (2012). *Pegaso: una propuesta para la gestión activos de software*. Tesis de Maestría, Universidad EAFIT.
- Bass, L.; Clements, P.; Cohen, S.; Northrop, L. y Withey, J (1997). *Product line practice workshop report*. Reporte técnico CMU/SEI-97-TR-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.
- Bass, L.; Clements, P.; Donohoe, P.; McGregor, J. & Northrop, L (2000). *Fourth product line practice workshop report (No. CMU/SEI-2000-TR-002)*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
- Concepción, J.R (2015). (10 de mayo de 2016). Acreditada la UCI como productora internacional de software, Cubadebate, Noticias, Ciencia y tecnología. Recuperado de: <http://www.cubadebate.cu/noticias/2015/10/19/acreditada-la-uci-como-productora-de-software-internacional-fotos/>.
- Espinosa, M. E. C., & Salavert, I. R (2016). *Ingeniería de aplicaciones dirigida por modelos y basada en técnicas de líneas de producción software*.
- Garlan, D., & Shaw, M (1994). *An introduction to software architecture*. Advances in software engineering and knowledge engineering, 1, 3-4.
- IEEE Computer Society (2010). *IEEE Standard for Information Technology – System and Software Life Cycle Processes - Reuse Processes*. IEEE Std 1517™. New York, USA: Society IEEE Computer.
- Miranda, M. F., Casas, S. I., & Marcos, C. A (2015). *Análisis de desarrollo de software orientado a Feature-Línea de producto de Software para Aplicaciones de TVDI*. Informes Científicos-Técnicos UNPA, 7(2), 167-195.
- Mustelier Sanchidrian, Daimara (2014). *Método para determinar la complejidad de los requisitos funcionales de software*. Tesis presentada en opción al título de Máster en Informática Aplicada, Universidad de las Ciencias Informáticas, La Habana.
- Ramos Blanco, K., Suárez Batista, A., Pérez Montalván, D., Neuland Agüero, D., Febles Estrada, A., Del-



- gado Martínez, R., & Muñoz Rojas, M (2011). Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas*, 5(2).
- Ramos Blanco, K (2013). *Proceso Base de Ingeniería de Requisitos para las pequeñas y medianas empresas de desarrollo de software*. Tesis en opción al título de Máster en Calidad de Software, Universidad de las Ciencias Informáticas, La Habana.
- Software Engineering Institute (SEI). (agosto de 2017). Carnegie Mellon University (2017a). Recuperado de: <https://www.sei.cmu.edu/productlines/casestudies/index.cfm>
- Software Engineering Institute (SEI). (22 de noviembre de 2017). Carnegie Mellon University (2017b). Recuperado de: <https://www.sei.cmu.edu/productlines/>.
- Stark, J (2015). *Product lifecycle management*. In *Product Lifecycle Management*. Springer International Publishing, 1, 1-29.
- Vázquez, M (2011). *Definición de una arquitectura de referencia para una línea de productos de software*. Tesis de Maestría, Universidad de las Ciencias Informáticas.
- Zubrow, D. & Chastek, G (2003). *Measures for software production lines*. Nota técnica CMU/SEI-2003-TN-031, Software Engineering Institute, Carnegie Mellon University.

