

# COTSRE+: Un método de desarrollo basado en componentes y requisitos<sup>1</sup>

María José Casalins Pina, Miguel A. Martínez Aguilar, Ambrosio Toval Álvarez

[mjcasalins@gmail.com](mailto:mjcasalins@gmail.com), [mmart@um.es](mailto:mmart@um.es), [atoval@um.es](mailto:atoval@um.es)

Grupo de Investigación de Ingeniería del Software. Departamento de Informática y Sistemas. Universidad de Murcia. Campus de Espinardo. 30071. Murcia. España

**Abstract.** En este artículo se presenta la definición de un método de desarrollo basado en componentes y requisitos, COTSRE+, en el cual se tiene en cuenta la reutilización. Se ha tomado como partida un método de selección de componentes a partir de requisitos y características, definido de forma inicial, llamado COTSRE. Se amplía la selección de componentes de COTSRE teniendo en cuenta los casos de uso además de los requisitos. Este método se complementa con una herramienta de escritorio que automatiza la selección de componentes de un repositorio llamada CotsreApp.

## 1. Introducción

Aunque el concepto de reutilización [1] no es nuevo, ha emergido como elemento central de éste, el concepto de componente software reutilizable. Las características de los componentes de software reutilizables han obligado a crear nuevos modelos de desarrollo de software basados en la reutilización. Otra de las deficiencias encontradas en los métodos basados en la reutilización es que no especifican de forma detallada el ciclo de vida de un componente de software reutilizable. Estos métodos se centran normalmente en la reutilización del componente y no en su desarrollo individual.

En este artículo presentamos la definición de un método de desarrollo basado en componentes y requisitos. En el capítulo 2 haremos una revisión del estado del arte y de los antecedentes a la hora de definir COTSRE+, como son el proceso de Chessman y Daniels [2] (en adelante C&D), algunos métodos de selección de componentes y COTSRE. En el capítulo 3, se presenta el método de desarrollo basado en componentes y requisitos que presentamos llamado COTSRE+. Finalmente el capítulo 4 muestra las conclusiones y el trabajo futuro.

---

<sup>1</sup> Trabajo parcialmente subvencionado por el Ministerio de Ciencia y Tecnología, proyecto PANGEA (*Process for globAl requiremeNts enGinEering and quAlity*, TIN2009-13718-C02-02), integrado en el proyecto PEGASO (*Processes for the Improvement of Global Software Development*, TIN2009-13718-C02-01).

## **2. Estado del arte**

En este apartado, presentamos una revisión de los principales trabajos de la literatura, relacionados con los métodos de selección de componentes, antes de definir COTSRE+. Para desarrollar el ciclo de vida de los componentes se ha utilizado una versión del proceso propuesto por C&D.

### **2.1. Proceso de Chessman y Daniels**

C&D definen un proceso de desarrollo basado en componentes el cuál se divide en 6 fases, de requisitos, especificación de componentes (a su vez dividida en identificación, interacción y especificación de componentes), aprovisionamiento de componentes, ensamblaje, pruebas e implantación.

En COTSRE+ hemos intentado mejorar este proceso, para ello hemos ampliado la fase de aprovisionamiento especificando cómo hacer la selección de componentes en caso de querer reutilizar alguno ya existente, la fase de pruebas la hemos dividido y adelantado para poder realizar pruebas unitarias de componentes y pruebas en el prototipo de la aplicación, además se han documentado todas las fases y se ha definido el conjunto de roles relacionados.

### **2.2. Métodos de selección de componentes**

Hemos revisado otros métodos para obtener una información sobre selección y la creación de componentes lo más amplia posible para definir COTSRE+. En la Tabla 1, se hace un resumen de los principales métodos de selección y sus características más importantes. Como se observa, el más completo para las características que creemos que debería cumplir un buen método de selección de componentes es SCARLET. A pesar de ello, hay dos características que creemos que son importantes en un buen método, cómo son su facilidad de uso o una buena negociación de requisitos, que no se cumplen para SCARLET. De aquí la necesidad de crear un nuevo método de selección de componentes ampliando y mejorando las fases contempladas en el método de C&D, ya que éstos no contemplan en su proceso ningún método específico para la selección de componentes.

### **2.3 COTSRE**

COTSRE+ parte de una versión preliminar de un método de selección de componentes llamado COTSRE (*Component Selection method based on Requirements Engineering*) [8, 9], el cual se basa en el método SIREN [10, 11, 12] y hace uso de estándares de la disciplina de Ingeniería del Software y de catálogos de requisitos reutilizables. Este método está definido de forma incompleta y solo para lo que sería la fase de selección de un componente de un determinado repositorio.

COTSRE aporta un proceso ágil para los desarrolladores del software a la hora de escoger un componente a partir de un conjunto de requisitos. Inicialmente se propone que cada requisito del catálogo de requisitos reutilizables contenga un atributo llamado “fuente”, que indique el lugar donde se puede encontrar un componente software (o varios) que lo cumplan. De esta manera, los desarrolladores buscarán los requisitos que necesitan dentro del catálogo de requisitos. Una vez tengan estos, mirarán el atributo “fuente” de cada uno en busca de los componentes que los cumplan. Con toda probabilidad no encontraremos sólo uno, sino un conjunto inicial de posibles candidatos. A la hora de seleccionar qué componente se adapta mejor al conjunto de requisitos necesarios juegan un papel muy importante los requisitos no funcionales como pueden ser el lenguaje de programación, el sistema operativo, la memoria disponible, el tiempo de ejecución, tipo de software (libre o propietario), etc. Estos requisitos imponen fuertes restricciones en la aplicación final a desarrollar, por lo que deben utilizarse a la hora de escoger el componente [13].

Características/Método	OTSO [3]	PORE [4]	CRE [5]	SCARLET [6]	CARE/SA [7]
Año de publicación	1995	1999	2001	2002	2004
Adquisición de requisitos	NO	SI	PARCIAL	SI	SI
Requisitos no funcionales	PARCIAL	PARCIAL	SI	SI	SI
Evaluación de componentes	SI	SI	SI	SI	SI
Análisis de toma de decisiones	SI	SI	SI	SI	SI
Basado en rechazo	NO	NO	NO	SI	NO
Selección múltiple	NO	NO	NO	SI	NO
Identificación de componentes	SI	SI	SI	SI	SI
Facilidad de uso	SI	PARCIAL	PARCIAL	NO	NO
Herramienta CARE	NO	NO	NO	SI	NO
Aceptación del componentes	SI	SI	SI	SI	PARCIAL
Téc. Análisis de Características	NO	SI	NO	SI	NO
Define métricas a utilizar	NO	NO	NO	NO	NO
Negociación de requisitos	NO	NO	NO	NO	PARCIAL

**Tabla 1.** Resumen técnicas de selección de componentes

Para ayudar en el proceso de selección se propone hacer una matriz de selección para tener una visión rápida de la relación entre los requisitos que son necesarios que cumpla la aplicación y los componentes seleccionados.

### 3. COTSRE+

Partiendo del proceso de C&D, hemos completado COTSRE para convertirlo en un método de desarrollo basado en componentes y requisitos bien definidos utilizando la notación de SPEM [14] y renombrándolo COTSRE+.

Se han definido una serie de roles que participan en COTSRE+ los cuales dividimos en dos grupos, los pertenecientes a la parte del cliente (usuarios finales y expertos en el dominio) y los de la parte técnica (ingeniero de requisitos, analista, seleccionador, testeador, programador y documentalista). Esta ha sido una de las principales aportaciones de COTSRE+ ya que C&D no definen bien cuáles son los roles existentes a lo largo de su proceso.

### **3.1. Fases de COTSRE+**

Vamos a definir los objetivos de cada una de las fases y detallando un poco más aquellas donde hagamos una aportación importante a C&D.

Cada fase es documentada al final de la misma para guardar un histórico del proceso de desarrollo. De esta forma, al finalizar el desarrollo de una aplicación utilizando COTSRE+, se tenga una documentación completa del mismo.

La fase de pruebas se ha eliminado realizando las pruebas individuales de cada componente en la fase de aprovisionamiento y las pruebas de la aplicación desarrollada en la fase de ensamblaje y despliegue.

#### **3.1.1. Fase de requisitos**

El objetivo de esta fase es obtener toda la información posible de los usuarios finales y expertos del dominio y detectar cuales son los requisitos y casos de uso del sistema. Esta fase es muy importante ya que a partir de toda esta información se podrá comenzar el proceso de análisis de los componentes de la aplicación. El documento generado al final de esta fase se utiliza como contrato con los clientes. Tanto para los casos de uso como para el documento se ofrecen unas plantillas para hacerlos.

#### **3.1.2. Fase de especificación de componentes**

Esta fase es dividida en otras tres:

- Fase de identificación de componentes: En esta fase se crea un conjunto inicial de interfaces y especificaciones de componente, conectadas entre sí a una primera aproximación de la arquitectura de componente. Lo que se pretende averiguar es que información necesitamos manejar, que interfaces la manejan, que componentes necesitamos para proveer que funcionalidad y como va a encajar todo esto.
- Fase de interacción de componentes: En esta fase se decide cómo van a interactuar los componentes identificados en la fase anterior. Usamos diagramas de interacción para definir las interacciones que tienen lugar dentro del sistema, para refinar las definiciones de las interfaces existentes, para identificar cómo son usadas las interfaces y para descubrir nuevas interfaces y operaciones.
- Fase de especificación de componentes: El objetivo de esta fase es crear los contratos de uso y realización. Los contratos de uso vienen definidos por la especificación de las interfaces y los de realización por la especificación de los componentes.

#### **3.1.3. Fase de aprovisionamiento y desarrollo**

El objetivo de esta fase, donde COTSRE+ hace su principal aportación, consiste en especificar los componentes según la tecnología escogida y seleccionar aquellos com-

ponentes que necesitamos para la aplicación. Además se desarrollarán aquellos componentes para los que no se ha encontrado un componente reutilizable según su especificación y todos ellos tanto los implementados como los reutilizados serán probados individualmente. Se utiliza una aplicación llamada CostreApp para ayudar en la selección de componentes por requisitos y/o casos de uso de un repositorio. Los casos de uso se buscan según el objetivo que desea que cumplan.

#### **3.1.4. Fase de ensamblamiento y despliegue**

En esta fase se realiza la implementación de la aplicación y parte de la fusión de las fases de ensamblaje y despliegue de C&D. Se realizaran pruebas de integración de los componentes además de las pruebas que los usuarios finales hacen sobre el funcionamiento y la interfaz visual. El objetivo de esta fase es la entrega final de la aplicación junto con un manual de utilización a los clientes. Además al término de esta fase se catalogan los componentes y requisitos para una futura reutilización.

## **4. Conclusiones y Trabajo Futuro**

Hemos presentado un método que completa y amplía otras propuestas existentes y que hace hincapié en la importancia de la reutilización de componentes. Partiendo de una especificación inicial de un método de selección de componentes, COTSRE, se ha generado un proceso que aborda todo el ciclo de vida de los componentes. Para ello, nos hemos basado en la propuesta de C&D ya que es un proceso que creemos bien estructurado y sencillo pero que no especificaba claramente cómo se debía llevar a cabo la selección de componentes. Al método final lo hemos renombrado como COTSRE+. El proceso de selección se completa teniendo en cuenta los casos de uso para seleccionar un componente. Incluso tiene en cuenta que no se encuentre ningún componente que cumpla con toda la especificación, en cuyo caso es capaz de relajar la condición para considerar aquellos candidatos que no cumplen exactamente con toda la especificación.

Aunque el método ha sido diseñado, basado y pensado en componentes, entendemos que el mismo podría ser adaptado sin grandes problemas en el ámbito del SOC (*Service Oriented Computing*), dada la similitud entre ambos procesos de desarrollo.

Somos conscientes que nuestra propuesta no deja de ser una propuesta teórica, que necesita ser aplicada y validada en situaciones reales para acabar obteniendo un método realmente práctico y útil. No obstante, es un primer paso, al que se ha dotado de la infraestructura de herramientas necesarias para mostrar también su viabilidad en este sentido.

Algunas posibles vías futuras del método COTSRE+ se detallan a continuación:

1. Validación de COTSRE+ mediante uno o varios casos de estudio.
2. Mejorar el método de selección de componentes por ejemplo mediante el uso de herramientas de búsqueda y análisis de similitudes entre texto y otras formas de representación [15, 16, 17].
3. Ampliar los catálogos de componentes reutilizables haciéndolos públicos.
4. Añadir una fase de mantenimiento al final del ciclo de vida de COTSRE+.

## Referencias

- [1] Mili, H., F. Mili, and A. Mili, Reusing Software: Issues and Research Directions. IEEE Transactions on Software Engineering, 1995. 21(6): p. 528-562.
- [2] John Cheesman y John Daniels. "UML Components. A Simple Process for Specifying Component-Based Software" Ed. Addison- Wesley, 2000.
- [3] Kontio, J. "A Case Study in Applying a Systematic Method for COTS Selection", 18th International Conference on Software Engineering, Springer (1996) 291-209.
- [4] Ncube, C., Maiden, N. "PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm". Proceedings of the 2nd International Workshop on Component Based Software Engineering (CBSE), Los Angeles, USA (1999).
- [5] Alves, C. Castro, J. "CRE: A systematic method for COTS component selection". Proceeding of the XV Brazilian Symposium of Software Engineering (SBES), Rio de Janeiro, Brazil (2001).
- [6] Maiden, N.A.M., Croce, V., Kim, H., Sajeve, G., Topuzidou, S. "SCARLET: Integrated Process and Tool Support for Selecting Software Components". Cechich, A. Piattini, M., Vallecillo, A., eds. "Component-Based Software Quality.
- [7] Cheng, L., Cooper, K., "COTS-Aware Requirements Engineering and Software Architecting", Software Engineering Research and Practice (2004) 57-63.
- [8] Miguel A. Martinez, Ambrosio Toval, Manuel F. Bertoa y Antonio Vallecillo. "Towards a Component Based and Requirements Driven Development". Handbook on Computer Software Engineering Research. Nova Science Publishers (ISBN: 978-1-60021-774-6).
- [9] Miguel A. Martinez, Ambrosio Toval, "COTSRE: a Component's Selection method based on Requirements Engineering", Universidad de Murcia. The 7th. IEEE Int. Conf. on Composition-Based Software. ICCBSS 2008 Student Session. February 25, 27, 2008, Madrid, Spain.
- [10] A. Toval, B. Moros, J. Nicolas, J. Lasheras. "Eight Key Issues for an Effective Reuse-Based Requirement Process", Computer Systems Science and Engineering. Vol. 23 (6). 2008.
- [11] A. Toval, A.Olmos, M.Piattini "Legal Requirements Reuse: A Critical Success Factor for Requirements Quality and Personal Data Protection" Proceedings of the IEEE Joint International Conference on Requirements Engineering (ICRE'02 and RE'02), pp: 9-13, Septiembre 2002, Essen Germany, IEEE Computer Press. Pp. 95-103. ISBN 0-7695-1465-0; ISSN 1090-705X.
- [12] A.Toval, J.Nicolás, B.Moros, F. García. "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach". Requirements Engineering Journal vol. 6, n.4, pp. 205-219, 2001.
- [13] Manuel F. Bertoa, José M. Troya, Antonio Vallecillo, "Aspectos de calidad en el desarrollo de software basado en componentes", Capítulo 8 del libro "Calidad en el desarrollo y mantenimiento del software", RA-MA, 2002. ISBN: 84-7897-544-6.
- [14] SPEM <http://www.omg.org/spec/SPEM/2.0/> Consultado 15 de Mayo 2010.
- [15] Maria Pierez, Sven Casteleyn, Ismael Sanz, Maria Jose Aramburu. "Requirements gathering in a model-based approach for the design of multi-similarity systems." In MoSE+DQS'09: Proceeding of the 1<sup>st</sup> international workshop on model driven service engineering and data quality and security, pages 45-52, New York, NY, USA, 2009. ACM.
- [16] The Apache Software Foundation, "Apache lucene" [http:// lucene.apache.org/java/2](http://lucene.apache.org/java/2) Consultado el 2 de Febrero de 2009.
- [17] K. Indukuri, A. Ambekar, A. Sureka, "Similarity analysis of patent claims using natural language processing techniques." International Conference on Computational Intelligence and Multimedia Applications, 2007, vol.4, pp. 169-175, Dec. 2007.