

# Análisis de los Métodos de Selección de Componentes COTS desde una Perspectiva Ágil

Fredy Navarrete, Pere Botella, Xavier Franch  
*Universitat Politècnica de Catalunya*  
*{fjnavarrete, botella, franch}@lsi.upc.edu*  
<http://www.lsi.upc.edu/~gessi>

## Resumen

Los métodos ágiles de construcción de software como eXtreme Programming y Scrum conciben el proceso de construcción como un ejercicio de desarrollo de software a medida. Sin embargo en la actualidad, muchos sistemas software están compuestos parcial o incluso totalmente por componentes COTS (Commercial Off-The-Shelf), debidamente personalizados e integrados. En este tipo de sistemas, el énfasis se desplaza de las actividades propias del desarrollo a medida, a los procesos de selección e integración de COTS. Existen diferentes propuestas basadas en la selección de COTS (pocas tratan de integración), y en este artículo estudiamos cómo se comportan desde un punto de vista ágil. Concretamente, introducimos los principios ágiles en el contexto de selección de COTS y analizamos bajo esta perspectiva algunas de las propuestas de selección de COTS más aceptadas. Como resultado, identificamos algunas prácticas que pueden dotar de mayor agilidad a los procesos de selección de COTS.

## 1. Introduction

En la actualidad, los métodos ágiles [1, 2] juegan un rol importante en las prácticas de la ingeniería del software. Métodos como eXtreme Programming (XP) [3] y Scrum [4], entre otros, han sido adoptados por organizaciones y equipos de desarrollo, que han informado de diversos resultados satisfactorios (asimismo, se han informado de experiencias no tan satisfactorias, como es habitual en todo tipo de método). Sin embargo, en nuestra opinión, el problema de los métodos ágiles es que tienden a centrarse únicamente en sistemas software desarrollados a

medida por un equipo de programadores, donde la reutilización esta limitada a repositorios de componentes de software, manejados básicamente por el mismo equipo (o por otra parte de la organización). Si por ejemplo consideramos las prácticas de XP, tales como la programación por pares o la simplicidad del diseño, puede ser complicado extrapolarlas a otro contexto que no sea el del desarrollo a medida. Sin embargo, esta perspectiva no tiene en cuenta una gran porción del mercado software, tales como los sistemas de información cooperativos o software de comunicación, donde la construcción de sistemas se basa en la adquisición e integración de componentes COTS (Commercial Off-The-Shelf).

La pregunta que aparece de inmediato es si los métodos ágiles pueden ser aplicados al contexto de la construcción de software basado en COTS. La contribución de nuestro artículo se enfoca en esta dirección. Hemos identificado dos estados en nuestra investigación: en la primera fase nos proponemos analizar la agilidad de los procesos locales relacionados con COTS, básicamente selección e integración, y luego nos centraremos en la agilidad del proceso global de construcción de sistemas basados en COTS. En este artículo nos hemos centrado en el proceso selección de COTS.

El resto del artículo esta estructurado como sigue. Primero identificamos los principios ágiles que pueden ser aplicados en el contexto de los COTS (sección 2). A continuación, analizamos algunos de los métodos de selección de COTS más aceptados en relación con dichos principios ágiles (sección 3). Finalmente, a partir de nuestras experiencias, sugerimos algunas prácticas que pueden mejorar la perspectiva ágil de los métodos de selección de COTS (sección 4). Finalmente, citamos el trabajo relacionado y las conclusiones de nuestro artículo (sección 5).

## 2. Los principios ágiles en el contexto de selección de componentes COTS

Tomamos como punto de partida los “Principles behind the Agile Manifesto” resumidos en [5], con el fin de adaptarlos al contexto de la selección de COTS. Tras una primera revisión de los 12 principios, descartamos aquéllos que o bien no se aplican al contexto que nos ocupa, o bien no se ven afectados por el mismo. Esos principios son el 1, 6, 7 y 11. Los principios 1 y 7 no son aplicables porque nos centramos únicamente en la selección de COTS y no en el ciclo total de desarrollo, mientras que los principios 6 y 11 no parecen estar influidos por el hecho de usar o no COTS.

A continuación examinamos el resto de los principios ágiles en el marco de la selección de COTS. Marcamos en *cursiva* las palabras que consideramos clave dentro de los principios.

**P1** “*Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage*”

En los proyectos COTS, los requisitos de los componentes COTS deben ser flexibles, para poder adaptarse a los constantes cambios y evolución del mercado. En [6], se menciona que el 31% de los proyectos estudiados apuntan a la necesidad de hacer flexibles los requisitos en la fase de definición. Los proyectos basados en COTS deben tener presente que los objetivos subyacentes de los *stakeholders* son más estables que los requisitos, que sufren modificaciones durante el desarrollo del proyecto [8], por eso debemos entender el “por qué” del proyecto (objetivos), por encima del “qué” de las características requeridas para seleccionar el componente COTS (requisitos) [7].

**P2** “***Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter time scale*”

Las iteraciones y el desarrollo incremental (IID) han servido de base a diferentes métodos donde el desarrollo de los procesos es iterativo [9]. La selección de componentes no es una excepción; la planificación de iteraciones dentro de los procesos COTS para realizar entregas frecuentes de resultados, servirá para mostrar avances ágiles dentro del proyecto, p. e., iterar para realizar exploraciones del mercado, analizar requisitos, evaluar COTS candidatos, integrar y mantener el sistema por completo, etc. Cabe destacar el

obstáculo que representan las dependencias entre COTS cuando la selección es múltiple, dificultando su análisis independiente.

**P3** “***Business people and developers must work together daily throughout the project.***”

En un proyecto COTS, la recomendación de los métodos ágiles de trabajar con el cliente debe ampliarse a la necesidad de trabajar conjuntamente con el proveedor de COTS, debido a la influencia que ejerce sobre la funcionalidad del COTS. Son estos proveedores quienes suministran un porcentaje alto de información de las características de los componentes, por esto debemos buscar beneficios mutuos para poder trabajar conjuntamente con el vendedor [6, 10], incluso pensar en incorporarlo al equipo de trabajo en la fase de integración del COTS. Además, debido a la naturaleza de este proceso, se hace indispensable incluir en el equipo de trabajo un rol capaz de manejar los aspectos contractuales y legales con el vendedor (ver sección 4).

**P4** “***Build projects around motivated individuals**. Give them the environment and support they need, and trust them to get the job done*”

En el contexto de los COTS, son pocos los métodos que consideran el factor humano como preponderante de sus procesos, soslayando pues aspectos humanos importantes como: el talento, la habilidad, cordialidad, sociabilidad del individuo y la comunicación [2], que impactan directamente en la cultura de la organización y en la gente que la compone, ya que son ellos quienes usan el sistema [11]. Además, son pocos los métodos que proponen roles específicos para conformar un equipo de trabajo, olvidando que en los proyectos de selección de COTS se generan nuevas responsabilidades y nuevas actividades, que deben ser cumplidas por nuevos roles que posean la habilidad y el conocimiento verificable.

**P5** “***Agile processes promote sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*”

Los procesos de selección de COTS no deben ser innecesariamente exhaustivos, con el fin de asegurar un flujo de trabajo que no debilite ni deteriore el ritmo interno del proyecto. Dos elementos juegan un papel clave. Por un lado, la arquitectura del sistema es la piedra angular en la que reposa todo el proceso de selección. El segundo aspecto es disponer de un repositorio de

información que almacene datos de diferentes fuentes de información (vendedores, componentes, requisitos y los procesos mismos).

**P6** “*Continuous attention to technical excellence and good design enhances agility.*”

“*High quality is the key to high speed*” [1]. En cada fase de los procesos de selección de COTS, la capacitación técnica de los miembros que componen el equipo de trabajo es vital para obtener un resultado de excelencia técnica. Para ello es necesario identificar roles de marcado perfil técnico (v. 4.1) capaces de manejar técnicas y artefactos que pueden mejorar la fiabilidad de los procesos de selección, tales como el modelado orientado a objetivos [7, 8], la negociación win-win [12] y la técnica AHP [13] entre otras. El problema de utilizar dichas técnicas radica en su complejidad, que puede requerir una formación especializada para su asimilación. Además, si para llevarlas a cabo se requiere una documentación excesiva, podemos interferir con el principio 7.

**P7** “*Simplicity--the art of maximizing the amount of work not done--is essential.*”

La simplicidad de los procesos de selección de COTS puede verse afectada por las técnicas, métodos y roles que hemos comentado en los otros principios, p. e., *P5*, *P6* y *P8* requieren de tiempo y esfuerzo para alcanzar ciertos objetivos en detrimento de la simplicidad. En el contexto de las metodologías ágiles, la clave para evitar que estos principios entren en conflicto radica en no anticiparnos al mañana. En el contexto de los COTS, la rápida evolución del mercado hace todavía más discutible invertir esfuerzos para anticiparnos al futuro.

**P8** “*At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly*”

Los proyectos de selección de COTS generan conocimiento y aprendizaje a medida que avanzan, que se convierten en una fuente importante de información en aras de ajustar el comportamiento dentro de la organización, para poder aprender del pasado. Este conocimiento debería estar almacenado en un repositorio. El repositorio debería almacenar información a base de feedbacks constantes de los resultados de los procesos de selección, integración y mantenimiento. Además, debemos incluir un rol en el equipo de trabajo responsable de mantener el repositorio (ver sección 4.1). Si los procesos para

almacenar esta información en el repositorio son lentos y están exageradamente documentados, afectaremos a *P7*.

### 3. Análisis ágil de los métodos de selección COTS

En nuestro trabajo hemos investigado 5 de los métodos de selección de COTS más aceptados: SCARLET [14], OTSO [15], CARE [16], PECA [17], STACE [11]. Debido a la falta de espacio, en esta sección analizamos en detalle los tres primeros métodos en relación a los 8 principios ágiles presentados en la sección 2. Para cada método, introducimos una tabla que relaciona sus aspectos más importantes respecto al principio respectivo, analizando si tiene un impacto positivo (+), negativo (-) o ambos (+-).

#### 3.1. SCARLET

El método SCARLET [14] (inicialmente conocido como BANCKSEC) adapta a su antecesor PORE [19] al dominio de la banca, permitiendo además la selección múltiple de componentes.

**P1.** Se reconoce la naturaleza cambiante de los requisitos, con el fin de definir procesos iterativos de adquisición de los mismos, durante los que se evalúan los COTS candidatos.

**P2.** En este método se descartan los COTS gradualmente, pero no se entregan resultados parciales cuando se tienen que seleccionar múltiples componentes. En [19], los autores proponen como trabajo futuro contar con un entorno *plug&play* del sistema para prototipado, pero no lo han llegado a implementar.

**P3.** Reconoce dentro de sus procesos la importancia de contar con los proveedores. Éstos influyen en los procesos de selección, en la elaboración del contrato al adquirir y asegurar la funcionalidad del componente, y en las pruebas de demostración del componente. Sin embargo, el proveedor no hace parte formal del equipo de trabajo sugerido por la metodología.

**P4.** No se tiene en cuenta el factor humano dentro de sus procesos, a pesar de que recomienda la configuración de un equipo de trabajo para llevar a cabo los procesos propuestos. Además, los humanos se consideran al mismo nivel que un componente software.

**P5.** Se proveen procesos guiados para los equipos de trabajo durante un proceso concurrente de selección, donde se determinan a la vez los requisitos de los stakeholders, la arquitectura del sistema y la selección de COTS candidatos.

**P6.** Integra métodos, artefactos y técnicas tales como AHP, Volere, etc. los cuales proveen un alto grado de excelencia técnica; el método maneja el dominio de la banca, y cuenta con una adquisición específica de requisitos, que hace que se cuente con procesos confiables para su entorno.

**P7.** Distingue tres tipos de plantillas a rellenar con diferente exigencia en cuanto al esfuerzo demandado: a medida que progresa la selección, es necesario realizar mayor esfuerzo de evaluación, pero al mismo tiempo quedan menos candidatos.

**P8.** Este método está básicamente orientado a selecciones aisladas, por ello no se propone un repositorio de datos, a excepción de [20] como trabajo futuro. Sin embargo, el método cuenta con una herramienta para soportar y almacenar las historias de evaluación, con el fin de proporcionar experiencia a los equipos de trabajo y aprender de procesos de evaluación pasados.

|                                 | P1 | P2 | P3  | P4 | P5 | P6 | P7 | P8 |
|---------------------------------|----|----|-----|----|----|----|----|----|
| Procesos solapados              | +  | +  |     |    |    |    |    |    |
| Diferentes tipos de plantillas  |    | +  |     |    |    |    | +  |    |
| No integra al vendedor          |    |    | -   |    | -  |    |    |    |
| Tiene técnicas como AHP, Volere |    |    |     |    |    | +  | -  |    |
| Procesos guiados                |    |    |     |    | +  |    |    |    |
| Sin Repositorio                 |    |    |     |    | -  |    | +  | -  |
| No define Roles                 |    |    |     | -  |    | -  |    |    |
| Procesos especializados         |    |    |     |    |    | +  |    |    |
| Maneja contratos y proveedores  |    |    | + - |    | +  |    |    |    |
| Considera la arquitectura       | -  |    |     |    | +  |    |    |    |

Tabla 1. SCARLET y los principios ágiles.

OTSO [15] puede ser considerado el primer método de selección de amplia aceptación. Propone los principios básicos que los métodos subsecuentes han ido incorporando.

**P1.** Tiene en cuenta que los requisitos pueden estar sujetos a cambios por parte de los *stakeholders*. Sin embargo, como resultado de procesos como el filtrado y entendimiento de los requisitos, el método sugiere crear un punto de referencia común para todas las alternativas (*baseline*), con el fin de contar con una base fundamentada para realizar la evaluación de costo y valor, el cual no cambia durante la ejecución del proyecto, lo que afecta la flexibilidad.

**P2.** Sugiere conducir actividades de búsqueda y descubrimiento en pequeños incrementos, y revisar la frecuencia de cada incremento.

**P3.** Reconoce la importancia de trabajar conjuntamente con el proveedor para obtener beneficios conjuntos. Sin embargo, el método no sugiere incluir al proveedor en el equipo de trabajo.

**P4.** No se centra en el factor humano, pero considera que para reutilizar satisfactoriamente el conocimiento adquirido a través del proyecto, no se depende exclusivamente de factores técnicos, también se depende de factores organizacionales, de motivación y de cuestiones legales;

**P5.** Promueve el desarrollo sostenible, con el fin de asegurar la reutilización a través de procesos bien fundamentados, apoyados de un repositorio.

**P6.** Define formalmente los criterios de evaluación, de manera que la evaluación de alternativas pueda realizarse con eficiencia y consistencia. Además, usa AHP para fundamentar el criterio de evaluación obteniendo un resultado más confiable.

**P7.** En esta propuesta se sugiere concentrar el esfuerzo en generar una clasificación sobre cada criterio de evaluación con el fin de no sobrepasar el tiempo estimado por cada evaluación, además, cuenta con una documentación sistemática de los resultados.

**P8.** El método se basa en la reutilización del conocimiento adquirido en el pasado. Es por esto que utiliza un repositorio para organizar el conocimiento obtenido, que se gestiona en la última fase definida en el método.

### 3.2. OTSO

|   | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
|---|----|----|----|----|----|----|----|----|
| Baseline fijo                                     | -  |    |    |    | +  |    |    |    |
| Pequeños incrementos                              |    | +  |    |    |    |    |    |    |
| Fase de filtrado                                  |    | +  |    |    |    |    | +  |    |
| No se tiene en cuenta al vendedor                 |    |    | -  | -  |    |    |    |    |
| Fases Concurrentes                                | +  | +  |    |    |    |    |    |    |
| Considera la madurez de la organización           |    |    |    | +  | +  |    |    | +  |
| Conduce eficientemente diferentes alternativas    |    |    |    | -  | +  |    |    |    |
| Repositorio de datos                              |    |    |    |    | +  |    | -  | +  |
| Considera la reutilización                        |    |    |    |    | +  |    | -  | +  |
| Tiene técnicas como AHP y la evaluación detallada |    |    |    |    |    | +  | -  |    |
| Fase de evaluación                                |    |    |    |    |    |    |    | +  |

Tabla 2. OTSO y los principios ágiles.

### 3.3. CARE

CARE [16] es un método orientado a objetivos y orientado a agentes.

**P1.** Los objetivos y los requisitos de los *stakeholders* pueden ser modificados durante el desarrollo de sus procesos. Además, los roles propuestos negocian los requisitos en diferentes etapas durante el desarrollo del sistema.

**P2.** Iterativamente realiza procesos de obtención, análisis, corrección y validación de objetivos, pero no está claro si se pueden realizar múltiples evaluaciones o selecciones frecuentes, con el fin de observar resultados tempranos.

**P3.** Aunque almacena y mantiene los datos de los vendedores en su repositorio, no tiene en cuenta la interacción en del vendedor en el equipo, o no destaca la importancia de trabajar con él.

**P4.** No da importancia al factor humano en sus procesos, considera al humano como un agente software o agente hardware, el cual posee intencionalidad. Aunque propone tres roles dentro de su método, no recomienda la confección de un equipo para el desarrollo de un proyecto COTS.

**P5.** Está organizado alrededor de la arquitectura, que provee un medio para el desa-

rollo sostenible. Está soportada por unos roles técnicos que interactúan entre sí secuencialmente.

**P6.** El método propone tres roles técnicos, los cuales deben aportar experiencia según el perfil del rol. Además, dentro de la propuesta se tienen en cuenta *frameworks* como el NFR y el *i\**, que aportan excelencia técnica.

**P7.** La documentación pesada con la que cuenta esta propuesta, y la complejidad de sus *frameworks*, pueden afectar la simplicidad.

**P8.** Cuenta con un repositorio como base de conocimiento, el cual almacena las características de los COTS, junto con sus objetivos y especificaciones. Sin embargo no propone un equipo de trabajo que pueda aprender de este conocimiento para ajustar su conducta.

### 3.4. Observaciones Finales

La tabla 4 resume el resultado de nuestro análisis. Identificamos los 8 principios utilizando el identificador  $P_i$  introducido en la sección 2. Por cada método  $M$  y principio  $P_i$ , clasificamos el grado de cobertura de  $M$  para  $P_i$  usando la siguiente lógica:

|  | P1 | P2  | P3  | P4  | P5 | P6 | P7 | P8 |
|--|----|-----|-----|-----|----|----|----|----|
| Negociación de objetivos                     | +  |     |     |     |    |    |    |    |
| Considera la arquitectura                    | -  |     |     |     | +  |    |    |    |
| Procesos iterativos                          |    | + - |     |     |    |    |    |    |
| Recupera datos de los vendedores             |    |     | +   |     |    |    |    |    |
| No interactúa con el vendedor                |    |     | -   |     | -  |    |    |    |
| Propone 3 roles técnicos                     |    |     | + - | + - |    | +  |    |    |
| Interacción lógica y secuencial de los roles |    |     |     |     | +  |    |    |    |
| Frameworks NFR, <i>i*</i>                    |    | -   |     |     |    | +  | -  |    |
| Repositorio de datos                         |    |     |     |     | +  |    | -  | +  |
| No considera Factores Humanos                |    |     | -   | -   |    |    |    |    |

Tabla 3. CARE vs. los principios ágiles.

- 1- 'a' significa que  $M$  cubre totalmente a  $P_i$ .
- 2- 'b' significa que aunque  $M$  no tenga intención explícita sobre  $P_i$ , lo maneja bien. También marcamos con 'b' si el principio se menciona pero no queda claro que se trata.
- 3- 'c' significa que  $M$  no trabaja bien con  $P_i$ , aunque exista una presencia parcial de  $P_i$  en  $M$ .
- 4- 'd' significa que  $P_i$  no tiene presencia en  $M$ .

|         | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
|---------|----|----|----|----|----|----|----|----|
| SCARLET | a  | c  | c  | c  | b  | a  | c  | b  |
| OTSO    | b  | a  | c  | d  | b  | b  | d  | b  |
| CARE    | a  | c  | c  | b  | a  | a  | d  | a  |

Tabla 4. Análisis comparativo de los métodos vistos.

Observamos que el principio P1 está bien cubierto por todos los métodos, probablemente porque los métodos seminales OTSO y PORE recogieron explícitamente la importancia de realizar concurrentemente la adquisición de requisitos y la evaluación de COTS. Lo mismo sucede con el principio P6, lo que puede ser consecuencia que la mayoría de los métodos propuestos provienen del mundo académico. De hecho, en [21] se ha reportado que las técnicas empleadas en estos métodos son rechazadas en el ámbito industrial, debido a la presiones del tiempo.

El principio P5 y P8 están razonablemente bien cubiertos por los métodos. El desarrollo sostenible proviene de procesos bien definidos, mientras la reflexión está soportada por los repositorios.

El resto de principios no están bien cubiertos por todos los métodos. Es difícil reconciliar la simplicidad con otros principios, tampoco es obvio que la selección de COTS pueda conducir entregas frecuentes. Por último, se nota que los métodos actuales no consideran las motivaciones individuales como factor clave de sus procesos, centrándose más en los aspectos técnicos (P4). Esta puede ser la causa de que P3 no esté bien cubierto por los métodos actuales, los cuales no consideran importantes los aspectos legales del proyecto, donde a menudo el resultado de la selección de COTS se define como un conjunto de productos, en vez de un pliego de contratos como en realidad es. Además, se desconoce el potencial que se puede obtener al trabajar con los proveedores conjuntamente, donde por lo general

el proveedor es visto como un adversario más que un colaborador.

#### 4. Mejora de la agilidad de los métodos de selección de COTS

Como parte final del trabajo llevado a cabo en las secciones anteriores, enumeramos un conjunto de "buenas prácticas" (*good practices*) que pueden mejorar la agilidad a los métodos de selección de COTS actuales y desarrollamos una agenda de investigación. Las prácticas surgen de nuestras experiencias industriales en el ámbito de selección de COTS (existen referencias para algunas de ellas [31, 32, 33]).

##### 4.1. Practicas propuestas

- Constituir un equipo multidisciplinario de COTS combinando roles técnicos y no-técnicos. Los roles técnicos deben incluir: *requirements engineer*, que sea capaz de obtener, analizar y definir los diferentes objetivos y requisitos de los stakeholders; *market watcher*, debe clasificar los tipos de productos disponibles en el mercado, identificando a su vez los cambios sustanciales que pueden impactar o influenciar sobre el sistema de información de la organización; *component screener*, debe ser capaz de buscar componentes candidatos que se adapten a los requisitos que necesiten de un análisis más detallado; *component evaluator*, posee un alto perfil técnico capaz de aplicar técnicas y procesos que permitan evaluar detalladamente los productos candidatos. También se debe contar con algunos roles clásicos, como el *quality engineer* y un *project manager*. Por ultimo, un *component customizer*, que provenga del lado del proveedor, capaz de personalizar el componente cada vez que sea necesario. Los roles no-técnicos deben incluir: *system client*, quien será el encargado de determinar y validar los requisitos; *COTS supplier*, provee información detallada de versiones parciales de los COTS durante el análisis; *manager*, el cual comparte y administra responsabilidades dentro del equipo técnico y la organización; *lawyer*, provee asistencia en el momento de redactar los contratos y estudiar la licencia y mantenimiento de los componentes.

- Mantener un repositorio del proyecto. Esta idea puede parecer contradictoria con la filosofía ágil,

pero esta centrada en proveer mejoras a los procesos, a base de aprender del conocimiento adquirido, reutilizando objetivos, descripciones, evaluaciones, etc., con lo que incide en los principios P5 y P8. La clave consiste en no incluir conocimiento innecesario y costoso de calcular. Se hace indispensable poder almacenar en este repositorio las decisiones que se han tomado a través del proyecto (por qué se rechaza un componente, por qué se descarta un proveedor, técnicas utilizadas para realizar la evaluación, etc.).

- La evaluación por pares. Es una extrapolación inmediata de la programación por pares al contexto de la selección de COTS, y se justifica con argumentos similares.

- Incluir en el proceso de selección una “metáfora del componente”. La idea de metáfora se extrae del mundo ágil [1], y permite ganar en entendimiento y percepción de lo que debe hacer el componente que deseamos seleccionar. Esta metáfora debe ser construida teniendo en cuenta los objetivos y requisitos claves del componente.

- *Call for tenders*. El *tendering* [22, 23] es un procedimiento que puede ser obligatorio en algunos contextos (p.e., para las administraciones públicas a partir de cierto presupuesto) que nosotros proponemos como norma general cuando el COTS a seleccionar es de alta importancia estratégica (es decir, cuya mala selección podría comprometer seriamente el futuro de la empresa). Este proceso implica el desarrollo de dos etapas, el antes y el después de la resolución del *tendering*. Realizar una oferta inicial a los proveedores implica recibir mucho *feedback* por parte de los proveedores, los cuales pueden complementar nuestra oferta inicial, aportando o descartando características del componente que nos puedan proveer. Además, obtenemos información que puede ser de provecho al momento de seleccionar un componente. Una variante de esta técnica es el uso de cuestionarios como complemento de obtención de información sobre los productos y los proveedores [24].

#### 4.2. Agenda de investigación

- Definir un modelo de madurez similar al CMM para los procesos de selección de COTS. Este modelo permitiría a las organizaciones reflexionar

sobre la calidad de sus procesos y planificar su mejora. Esta idea ha sido explorada en [25].

- Proponer nuevos modelos de negocios. Actualmente existen organizaciones con ánimo de lucro y sin ánimo de lucro, al igual que compañías que actúan como intermediarias, ofreciendo catálogos extensos de productos COTS, que recogen la descripción de COTS, etc. [26, 27]. Los modelos de negocio actuales pueden determinar nuevas prácticas que no han sido tenidas en cuenta.

- Construir ambientes *plug&play* que simulen la arquitectura del sistema, con el fin de simular la integración del componente a la arquitectura, y razonar sobre su comportamiento. Esta idea fue propuesta en [19] y elaborada en un contexto orientado a objetivos [28].

## 5. Conclusiones

En este trabajo, hemos analizado los métodos actuales de selección de COTS bajo la perspectiva de los principios ágiles. Hemos identificado que algunas características de estos métodos se encuentran influenciadas positiva o negativamente por los principios ágiles, y hemos identificado algunas prácticas que pueden mejorar y proveer agilidad a los métodos de selección de COTS actuales. Además, hemos desarrollado una agenda de investigación con 3 puntos claves. Muchas de nuestras observaciones y prácticas son el resultado de diferentes trabajos [6, 10, 18, 19] los cuales pueden ser considerados como validación preliminar de nuestro trabajo, quedando pendiente de una implantación para una validación real que haga parte de un trabajo futuro.

Hasta la fecha, no existen muchos trabajos que se centren en el análisis de la agilidad de los métodos de selección de COTS. En [29] se trata el contexto de la implementación de un sistema ERP, el cual está enfocado más en la gestión del proyecto y en la implementación que en la selección. Algunas de las prácticas y heurísticas ágiles identificadas (pero no todas) pueden ser extrapoladas al contexto general de la selección de COTS.

Otra línea de investigación relacionada con nuestro trabajo es la adaptación de procesos de desarrollo existentes a los sistemas basados en COTS. En [30], el método RUP es analizado desde la perspectiva de los COTS, encontrando

similitudes frente a los métodos de selección de COTS, tales como la definición de roles específicos y el planeamiento de las iteraciones.

Sobre nuestro trabajo futuro, además de la validación (ver arriba), y tal y como mencionamos en la introducción, nuestro objetivo es extender este análisis al resto de tareas propias de la construcción de software basada en COTS, por ejemplo la integración, con el fin de obtener conclusiones sobre la agilidad de un ciclo completo de un sistema basado en COTS. El objetivo final es diseñar un nuevo método de construcción de software basado en COTS fundamentado en los principios ágiles, que sea adaptable según el tipo de organización.

## Referencias

- [1] Martin, R. C. *Agile Development: Principles, Patterns and Process*, Prentice Hall, 2002.
- [2] Cockburn, A., Highsmith, J. "Agile Software Development: The People Factor". IEEE Computer, December 2002.
- [3] Beck, K. *Extreme Programming Explained: Embrace Change*. Addison Wesley, 1999.
- [4] M. Beedle, K. Schwaber. *Agile Software Development with SCRUM*. Prentice Hall, 2001.
- [5] Beck, K. *et al. Manifesto for Agile Software Development*. <http://www.agilemanifesto.org>, 2001.
- [6] FAA SERC. "Lessons Learned in Developing Commercial Off-The-Shelf (COTS) Intensive Software System". October 2, 2000.
- [7] Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering". Proceedings of the 3<sup>rd</sup> IEEE ISRE, 1997.
- [8] Lamsweerde, A. "Goal-Oriented Requirements Engineering: A Guided Tour". Proceedings of the 5<sup>th</sup> IEEE ISRE, 2001.
- [9] Larman, C., Basili, V. "Iterative and Incremental Development: A Brief History". IEEE Computer, November 2004.
- [10] Brownsword, L., Place, P. "Lessons Learned Applying Commercial Off-the-Shelf Products". Report CMU/SEI-99-TN-015, June 2000.
- [11] Kunda, D., Brooks, L. "Applying Social-Technical Approach for COTS Selection". Proceedings of the fourth UKAIS Conference, University of York, April 1999.
- [12] Egyed, A., Kwan, J., Madachy, R. "Developing Multimedia Applications with the WinWin Spiral Model". In Proceedings ESEC/FSE 97, November 1997
- [13] Saaty, T.L. "How to make a decision: The analytic hierarchy process". European Journal of Operations Research, no. 48, pp. 9 - 26, 1990.
- [14] Maiden, N., Kim, H., Ncube, C. "Rethinking Process Guidance for Selecting Software Components". Proceedings of 1<sup>st</sup> ICCBSS, LNCS 2255, 2002.
- [15] Kontio, J. "A Case Study in Applying a Systematic Method for COTS Selection". In Proceedings 18<sup>th</sup> Intl' Conference on Software Engineering (ICSE), 1996.
- [16] Chung, L., Cooper, K., Courtney, S. "COTS-Aware Requirements Engineering and Software Architecting". Proceedings of the Intl' Conference on Software Engineering Research and Practice (SERP), 2004.
- [17] Dorda, C., Dean, C., Morris, E., Oberndorf, P. "A Process for COTS Software Product Evaluation.". Proceedings of 1<sup>st</sup> ICCBSS, LNCS 2255, 2002.
- [18] M. Morisio, C.B. Reaman, V.R. Basili, A.T. Parra, S.E. Kraft, S.E. Condon. "COTS-based software development: processes and open issues". Journal of Systems and Software 61 (2002): 189-199.
- [19] Maiden, N., Ncube, C. "Acquiring COTS Software Selection Requirements." IEEE Software 15(2), 1998.
- [20] Ncube, C., Maiden, N. "PORE: Procurement Oriented Requirements Engineering Method for a Component-Based System Engineering Development Paradigm." Proceedings of the 2nd International Workshop on Component-Based Software Engineering (CBSE), 1999.
- [21] Torchiano, M., Morisio, M. "Overlooked Aspects of COTS-Based Development". IEEE Software 21(2), 2004.
- [22] Lauesen, L. "Experiences from a tender process". Proceedings of REFSQ'04, Riga.
- [23] Krystkowiak, M., Bucciarelli, B., Dubois, E. "COTS Selection for SMEs: a report on a case study and on a supporting tool". Proceedings of the 1<sup>st</sup> RECOTS Workshop, September 2003.



- [24] Ncube, C., Maiden, N. "Selecting COTS Anti-Virus Software for an International Bank: Some Lessons Learned". *Proceedings 1st MPEC Workshop*, 2004.
- [25] Olson, T. "Using CMMI/SS to Manage COTS&MOTS Software". *Proceedings 2<sup>nd</sup> Annual CMMI Technology Conference and User Group*, 2002.
- [26] Mielnik, J.-C., Lang, B., Laurière, S., Schlosser, J.-G., Bouthors, V. "eCots Platform: An Inter-Industrial Initiative for COTS-Related Information Sharing". *Proceedings of 2<sup>nd</sup> ICCBSS, LNCS 2580*, 2003.
- [27] ComponentSource, <http://www.componentsource.com> last accessed Feb. 2005.
- [28] Franch, X., Maiden, N. "Modelling Component Dependencies to Inform Their Selection." *Proceedings of 2<sup>nd</sup> ICCBSS, LNCS 2580*, 2003.
- [29] Alleman, G.B. "Agile Project Management Methods for ERP: How to Apply Agile Processes to Complex COTS Projects and Live to Tell About It". In *Extreme Programming and Agile Methods: XP/Agile Universe LNCS 2418*, 2002.
- [30] Chan, R. "Adopting RUP in a COTS-Implementation Project". *The Rational Edge*, May 2003.
- [31] Franch, X., Carvallo, J. "Using Quality Models in Software Package Selection". *IEEE Software* 20(1): 34-41 (2003).
- [32] Carvallo, J., Franch, X., Quer, C., Rodríguez, N. "A Framework for Selecting Workflow Tools in the Context of Composite Information Systems". *DEXA 2004*: 109-119.
- [33] Carvallo, J., Franch, X., Quer, C. "A Quality Model for Requirements Management Tools". *Requirement Engineering for Sociotechnical System Chapter VIII* Pag 119 – 137.