



COMPARTIR *SOFTWARE* DE CÓDIGO ABIERTO



Estudio de viabilidad IDA
Intercambio de Datos entre
Administraciones
Comisión Europea, DG Empresa

Autores:

Patrice-Emmanuel SCHMITZ

Sébastien CASTIAUX

UNISYS

Junio de 2002

© Comunidades Europeas

Se autoriza la reproducción con indicación de la fuente.

El programa IDA

IDA (<http://europa.eu.int/ispo/ida>) es una iniciativa estratégica conducida por la Comisión Europea, en la que los progresos de las tecnologías de la información y la comunicación se utilizan para contribuir a un intercambio rápido de información entre las administraciones de los Estados miembros. El objetivo es mejorar el proceso de decisión de la Comunidad, facilitar el funcionamiento del mercado interior y acelerar la aplicación de las políticas.

Su [misión](#) es coordinar la creación de redes telemáticas transeuropeas:

fomentando la implementación de redes sectoriales en áreas prioritarias,
desarrollando medidas de interoperabilidad entre redes;
haciendo que la industria y los ciudadanos de la UE se beneficien de las ventajas de la red;
cooperando con las autoridades de los Estados miembros y los servicios de la Comunidad;
fomentando la convergencia hacia una interfaz telemática común.

Si desea hacer comentarios sobre este informe o asuntos relacionados con él, envíe un mensaje electrónico a la siguiente dirección:
ida-central@cec.eu.int.

Este informe ha sido elaborado bajo la responsabilidad exclusiva del contratista (*Unisys Management Consulting Team*) y no refleja necesariamente las opiniones de la Comisión Europea. La Comisión Europea tampoco acepta responsabilidad alguna respecto a la precisión o exhaustividad de la información del informe.

Si desea más información, póngase en contacto con el autor:
Patrice-emmanuel.Schmitz@be.unisys.com

Índice

Resumen

 ¿Compartir *software* de código abierto?

 Resumen del estudio

Resumen

¿Compartir *software* de código abierto?

Estudio de viabilidad

Este estudio es un estudio de viabilidad sobre cómo compartir (o poner en común o intercambiar) *software* y conocimientos entre las administraciones del sector público en toda Europa.

Para que el *software* del sector público se pueda reutilizar en mayor medida es necesario que sea *software* de código abierto. Con el fin de fomentar el uso compartido de *software*, vamos a examinar la viabilidad de crear un fondo común de *software*, lo que constituiría un servicio en el que se combinaran un centro europeo de conocimientos y un repertorio de mejores prácticas. En el presente estudio también se evalúan las limitaciones jurídicas, funcionales y técnicas de este proyecto.

Concepto de SCA (*software* de código abierto)

Se llama «*software* de código abierto» aquel en que el autor («cedente de la licencia») cede una serie de libertades básicas al usuario («licenciatario») en el marco de un acuerdo de concesión de licencia:

- La libertad de estudiar cómo funciona el programa y de adaptar su código a necesidades específicas; para ello es necesario poder acceder al código fuente.
- La libertad de mejorar el programa (ampliarlo, añadir funciones).
- La libertad de utilizar el programa con cualquier fin en cuantos ordenadores se desee.
- La libertad de proporcionar copias a otros usuarios.

¿Qué *software* vamos a examinar, y por qué de «código abierto»?

Todo el mundo conoce ejemplos de *software* de código abierto (SCA)¹, pero este estudio no se va a centrar en esos programas «estrella». El objeto del estudio es el *software* especializado producido por las autoridades públicas de toda Europa para responder a las necesidades de la administración o, más en general, de la administración pública en línea: gestión de carreteras, hospitales y salud pública, educación, pago y recaudación de impuestos, justicia, ordenación territorial².

Entonces, ¿por qué *software* de «código abierto»? Porque, por lo general, el *software* producido por las administraciones no son «paquetes comerciales», que pueden ser utilizados «tal cual» por otros usuarios. Además, Europa es un área de diversidad (idiomática, normativa, cultural, etc.) y un *software* desarrollado, por ejemplo, en Francia no se podrá utilizar en el Reino Unido o en Suecia: el deber de la administración francesa es satisfacer las necesidades de los ciudadanos de su país y no hacer dinero con un producto de tipo general que se pueda vender listo para usar. Por consiguiente, la reutilización de este tipo de *software* está supeditada a que se revise y adapte su código fuente.

De estos dos requisitos —la suposición de que no hay intereses comerciales en cuanto a derechos de licencia y la necesidad de facilitar el *software* junto con su código para adaptarlo a las circunstancias locales antes de implementarlo y redistribuirlo— se deriva naturalmente la idea de adoptar el «modelo de código abierto».

Por eso, el fin de este estudio no es examinar las ventajas o desventajas del *software* de código abierto y del *software* propietario, ni recomendar un sistema operativo o una aplicación dados, ya que el *software* desarrollado por las administraciones funciona en diversas plataformas (MS/Windows, Unix propietario, Linux, etc.). Tampoco se pretende tomar postura en el conflicto comercial —y, a veces, ideológico— entre los que abogan por la distribución gratuita de *software* y los que abogan por dar mayor protección a la propiedad intelectual e industrial del *software*.

Sólo se quiere examinar los requisitos y las condiciones (funcionales, jurídicas, técnicas) necesarias para un servicio paneuropeo de puesta en común.

Las ideas que subyacen al fondo común

¹ El más famoso SCA es, con mucho, el sistema operativo Linux (o, más exactamente, el núcleo Linux del «sistema operativo GNU/Linux»), pero sólo es el buque insignia de una numerosa flota con representantes más o menos ilustres.

² Ésta es una selección de las mejores prácticas presentadas en la conferencia de la administración en línea (Bruselas, 29 de noviembre de 2001).

Las ideas que subyacen al fondo común de *software* se clasifican en tres categorías: economía, calidad, filosofía.

Economía

- Al adquirir *software* ya existente para nuevas aplicaciones (perspectiva e intereses del licenciatario): gastar el dinero del contribuyente de la mejor forma posible reutilizando y adaptando *software* de mejores prácticas hecho por otras administraciones.
- Al reducir los costes de mantenimiento (perspectiva e intereses del cedente de la licencia): las aplicaciones ya existentes son costosas y difíciles de mantener y desarrollar de conformidad con nuevas normas. «Dar» su *software* en código abierto a la comunidad de desarrolladores puede suponer la gratuidad del mantenimiento y de nuevas versiones.
- Al compartir nuevos desarrollos de *software* cuando no existe previamente una solución (perspectiva común): en vez de desarrollar soluciones casi idénticas por separado, ¿por qué no adoptar el modelo de desarrollo de código abierto para repartir los costes entre un equipo de desarrollo más amplio (transfronterizo)?

Calidad

El objetivo de compartir no es necesariamente gastar menos, sino conseguir una calidad mayor por el mismo importe:

- Acelerar la innovación utilizando fondos para desarrollar aplicaciones realmente nuevas en vez de reinventar partes que ya han sido desarrolladas por otros.
- Permitir que unos países se beneficien del avance de otros: estudios comparativos han demostrado que hay «países punteros», que han hecho avances significativos en una o más áreas específicas.

Filosofía

- Fomentar la colaboración entre las administraciones europeas.
- Optimizar la cooperación con el sector privado concentrando la inversión en sectores realmente innovadores y fomentando nuevos servicios de asistencia.

Resumen del estudio

El estudio está organizado en seis capítulos, que corresponden a seis preguntas:

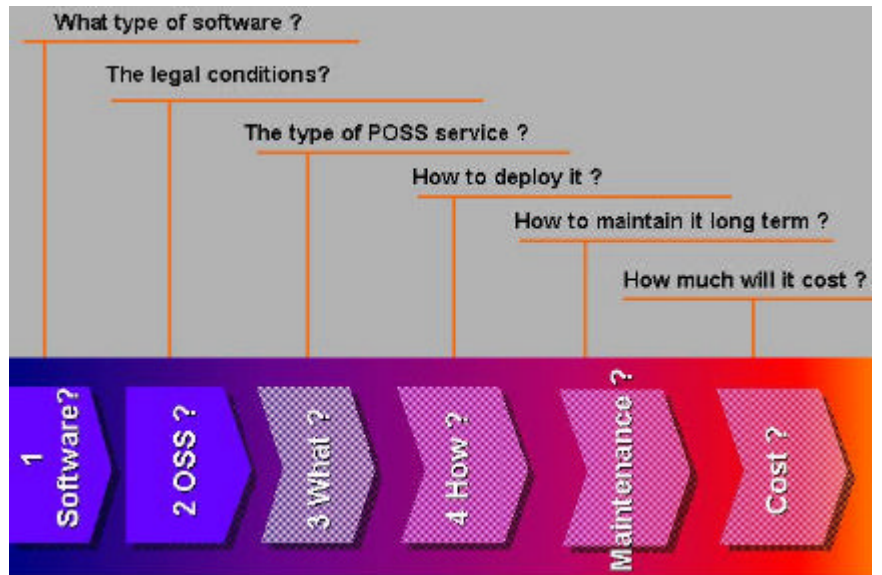


Figura 1: proceso del estudio CSCA

1. «Análisis de procesos y de recursos»

La pregunta «¿Qué tipo de *software*?» es pertinente porque, habitualmente, un requisito del código abierto es que se desarrolle en común: los programadores (empezando por una sola persona hasta llegar a formar una comunidad de desarrolladores) han desarrollado una solución interesante y que funciona para satisfacer sus propias necesidades o problemas (porque no había soluciones disponibles para su entorno o porque las soluciones existentes eran demasiado caras).

La pregunta a la que hay que responder es: «¿En qué medida o en qué condiciones el *software* desarrollado por una administración corresponde al modelo de desarrollo de código abierto?» ¿Proporciona este *software* una buena base para el desarrollo de código abierto? Para responder a estas preguntas, en este capítulo se analizan los procesos de desarrollo, adquisición e implantación de *software*, así como los recursos necesarios (personal, organización, proyectos, financiación, canales de distribución e información).

La conclusión principal es que compartir el *software* existente como código abierto para beneficiarse de mantenimiento y actualizaciones gratuitos es una ilusión si el *software* no se ajusta a condiciones estrictas de desarrollo:

- Número razonables de personas «que compartan el mismo problema».
- Reparto inicial pero flexible de la «propiedad / liderazgo» entre distintas personas de distintas organizaciones.

- Documentación de todo.
- Mapa de carreteras (poder navegar en el código como en un sitio *web*).
- Tronco común fácilmente comprensible y módulos funcionales, el código no puede ser monolítico.
- *Software* organizado en muchas porciones de código relativamente pequeñas (para facilitar la propiedad individual).
- Identificar (y declarar) claramente qué partes son «maduras / estables» y qué partes hay que «mejorar» (de acuerdo con el principio de «versiones frecuentes»).
- Crear foros de debate permanentes sobre los requisitos, objetivos y prioridades del desarrollo futuro.

Teniendo en cuenta todas estas condiciones, el hecho de «lanzarse al código abierto» puede suponer para el sector público una inversión inicial (*software* ya existente) o un considerable aumento de los costes de desarrollo (*software* nuevo) que se recuperarán a largo plazo.

2. Análisis del marco jurídico

Antes de compartir el código abierto con una comunidad, hay que definir las condiciones en que se hará y quién constituirá dicha comunidad: ¿otras administraciones, todo el mundo? ¿Se facilitará también el *software* a las empresas para fines comerciales? Por eso hay que analizar el marco jurídico para definir los papeles de proveedor, usuario y servicio intermediario, sus responsabilidades y los límites entre ellos, así como para orientar la elección de las condiciones contractuales que regirán el proceso: licencias y otras disposiciones.

La conclusión es que el marco contractual no se limita a las licencias de *software*, sino que incluye también:

- Las condiciones generales del servicio de puesta en común (lo que podríamos llamar «la carta CSCA»).
- El contrato entre el autor y el servicio CSCA (lo que podríamos llamar «mandato» o «encargo», ya que el servicio CSCA representará al autor —cedente de la licencia— en el contrato con el usuario —licenciataria—).
- Acuerdos específicos sobre responsabilidad, competencia jurisdiccional, legislación aplicable y patentes.
- El acuerdo de licencia en sí mismo debería ser seleccionado por el autor —cedente de la licencia— (y aceptado por el usuario —licenciataria—).

Por lo que se refiere a los beneficiarios del servicio CSCA, la conclusión común es que limitar el acceso al *software* compartido únicamente a las administraciones no es una preocupación principal pero sí debería estar limitado a usuarios registrados (principalmente el sector público, en un sentido amplio). La limitación no debería aplicarse al usuario final como tal y, por lo tanto, no se debería incluir en la licencia: no es posible hacerlo por motivos jurídicos (ya veremos que, debido al efecto «*copyleft*», las licencias SCA originales —como GPL y MPL— se deben usar sin modificaciones en determinadas condiciones) pero también por motivos prácticos: en un entorno de código abierto que permita la redistribución, es imposible controlar al usuario final.

Por eso, la limitación debería aplicarse al acceso al servicio de descarga de CSCA y a los requisitos de autenticación necesarios para que un contrato sea válido: sólo los usuarios (¿europeos?) registrados (administración pública, desarrolladores que trabajen para ella, socios industriales) tendrían que registrarse para poder bajarse programas. En cambio, no se controlará si los licenciarios iniciales redistribuyen luego los programas (con lo que dicha redistribución quedaría fuera del alcance y de la responsabilidad de CSCA, si la hubiera).

Por lo que se refiere a las distintas licencias de código abierto, están todas basadas en los derechos de autor pero, aunque se considera que cumplen las Convenciones sobre derecho de autor y sus distintas aplicaciones nacionales europeas, sus efectos varían muchísimo.

Hay más de 30 tipos o variantes distintas de licencias de SCA, por lo que CSCA debería orientar al posible cedente de la licencia con tres opciones: BSD, GPL y una variante de MPL³.

³ Se da por supuesto que el cedente de la licencia es libre de seleccionar la licencia que quiere y no está ya comprometido por condiciones de «*copyleft*»; es sabido, además, que el autor inicial es libre de usar varias licencias. Para los detalles, véase el capítulo 2.

La variante de la licencia MPL representa una interesante solución de compromiso, en la que se pueden separar el código y los ficheros binarios ejecutables: el código seguirá estando abierto (efecto «*copyleft*») y, además, se informará al autor original de todas las mejoras (importante para controlar y aprovechar las actualizaciones AE, AEX, etc.). Por el contrario, el objeto binario se *puede* distribuir con una licencia propietaria (que prohíbe la redistribución a través de una sencilla reproducción del medio por parte del usuario final). Además de los intereses del autor, esta modalidad protege los intereses comerciales de posibles distribuidores de obras derivadas (pueden exigir un precio justo al mismo tiempo que combaten la piratería) y los intereses de futuros desarrolladores: si necesitan añadir mejoras técnicas, pueden acceder al código, modificarlo, recopilarlo y, luego, redistribuir una nueva versión «original» (código y archivo binario).

Todo esto se expondrá únicamente con carácter informativo, puesto que la elección de la licencia será responsabilidad exclusiva del cedente.

Por último, se examinó el riesgo de que se produzcan solapamientos con el sector privado, pero dado el carácter específico del *software* en cuestión (desarrollado específicamente por la administración o para ella), se consideró que dicho riesgo es limitado: en el área CSCA, las aplicaciones e infraestructuras de código abierto abrirán un nuevo mercado para empresas de servicios y proveedores de soluciones, al tiempo que facilitarán la mejora de la calidad para llegar a las mejores prácticas europeas. El servicio CSCA no cuestiona el presupuesto global de desarrollo informático y la parte de dicho presupuesto que se externalizará para desarrollos del sector privado.

3. Análisis de requisitos funcionales

El capítulo 3 del estudio se elaboró sobre la base de los resultados de nuestro cuestionario: ¿Qué expectativas tienen las administraciones del sector público en cuanto al tipo de servicio CSCA?

Las principales conclusiones son las siguientes:

- Muchos proyectos desarrollados podrían usarse en otras administraciones (intercambio de datos, programas de grupo, sistemas de gestión de recursos humanos con interfaz *web*, etc.) pero, en la actualidad, hay una carencia de medios para proceder al intercambio.
- Un portal europeo CSCA debería tener en cuenta los sitios similares que ya se han desarrollado a nivel nacional, por ejemplo, facilitando enlaces con sus sitios.
- En el sitio debería haber una guía de referencia rápida para seleccionar el modelo de licencia.

- En el portal CSCA habría que definir las normas (codificación, formato de datos, etc.) que tendrían que cumplir todos los proyectos registrados en el sitio.
- Hay que examinar en detalle las cuestiones de seguridad.
- Estar en el sitio tiene que ser una cierta garantía (etiqueta) de la calidad del *software* registrado y, por lo tanto, hay que reflejar las apreciaciones de los usuarios.
- Limitar el acceso a CSCA únicamente a las administraciones no es algo que les preocupe en demasiado.

Para satisfacer estas necesidades, CSCA tendría que incluir una serie de servicios:

- Portal multilingüe con un mapa del sitio claro, contenidos paneuropeos y datos en formato normalizado.
- Gestión de los miembros registrados.
- Gestión del envío, descripción, clasificación y descarga de *software*.
- Enlaces con las iniciativas nacionales y cualquier otro tipo de enlaces relacionados con los proyectos (*software*) y los miembros.
- Gestión de bibliotecas para adjuntar diversos ficheros de documentación relacionados con el *software* (documentación técnica, manuales del usuario, etc.).
- Novedades, foros y listas de correo.
- Encuestas de opinión.
- Depósito de *software* para descarga (FTP) directa a partir de CSCA.
- Un comité científico. Aunque no se pueda proporcionar una garantía formal o una auditoría de calidad, una necesidad frecuentemente expresada es la conformidad con criterios formales de calidad (documentación, manuales, etc.) y la disponibilidad de protocolos de ensayo, así como el asesoramiento de expertos.
- Servicio de asesoramiento jurídico y gestión del marco contractual (incluyendo el registro de todas las descargas contractuales llevadas a cabo por usuarios autenticados, el *software* descargado, la fecha y la hora y el tipo de licencia acordado).

4. Marco para el diseño técnico

En el capítulo 4 del estudio se examina cómo habría que construir e implantar CSCA con arreglo a un marco de normas técnicas y se proporcionan ejemplos de posibles marcos de este tipo (servicios similares y recomendaciones).

Se tratan los siguientes temas:

- Definición del marco de normas técnicas que deben respetarse al diseñar CSCA. En esta parte se exponen el diseño general del sitio y los elementos de seguridad.
- Lista de las posibles herramientas necesarias para configurar CSCA. La lista se basa en las herramientas usadas por sitios similares y que han demostrado ser compatibles (sistema operativo, servidores *web* y FTP, base de datos, correo electrónico, estadísticas, controlador de enlaces, lenguaje de *scripts*). Las herramientas que se presentan como *ejemplo* son «**de código abierto**» (principalmente porque existen herramientas maduras de SCA y para evitar confusiones y discusiones que perjudicarían la credibilidad del estudio). No obstante, hay que decir que, para compartir *software* específico del sector público con licencias de código abierto, se podría hacer también —técnicamente hablando— utilizando uno o más componentes propietarios si lo justifican su coherencia, integración global y costes de entrega o rendimiento.
- Las distintas tareas que habría que realizar para configurar CSCA y la identificación de responsabilidades de cada tarea.
- Ejemplos de herramientas parecidas que ya existen y sus componentes.

5. Análisis de mantenimiento e interacción

Una cosa es construir una sola vez un servicio o un portal y otra su mantenimiento, que puede representar la mayor parte de la inversión a largo plazo.

En el capítulo 5 se dan respuestas a la pregunta de cómo mantener CSCA mediante la identificación de roles y la definición de los procesos de mantenimiento.

Los roles del servicio identificados son:

- El administrador del sistema
- El evaluador (del *software*)
- El controlador (del contenido diario de CSCA, usuarios, novedades, foros)
- El responsable del *software* (o del proyecto)
- El desarrollador
- El usuario

Los procesos de mantenimiento incluyen:

- **Componentes**
Equipo físico y *software* CSCA, así como el servidor *web*, la herramienta de foros, la infraestructura de correo electrónico y la herramienta para controlar enlaces.

- **Contenido**
Mantenimiento del índice de *software* y de las páginas iniciales, enlaces, novedades e índices de bibliotecas.
- **Servicios**
Registro de miembros, asesoramiento y marco jurídicos, foros, listas de correo, encuestas, estadísticas.

También se identifican las responsabilidades inherentes a cada proceso de mantenimiento.

A continuación se describe el problema de la integridad de los datos junto con la solución propuesta para el sitio CSCA.

Por último, se evalúa la posible contribución de organismos europeos.

6. Análisis de costes y de las opciones de financiación

El último paso en el estudio de viabilidad es reflexionar sobre los costes: costes para crear el prototipo (análisis y desarrollo), costes de implantación —que exigen esfuerzos considerables, no sólo técnicos sino también de información—, mantenimiento y funcionamiento, reevaluación periódica, actualizaciones y ciclo de vida de la aplicación CSCA.

La primera elección política se refiere a la propia noción de CSCA:

¿Por qué no empezar «sin costes» con un proyecto de *software* de código abierto, bajo la férula de uno o más gurús y con un ejército de voluntarios, que trabajen todos ellos gratuitamente en su tiempo libre para construir CSCA?

Nuestra respuesta es «No»: CSCA es **en primer lugar un servicio** que hay que prestar de forma constante; **no es sólo un *software*** que un grupo de voluntarios puede mejorar sin limitaciones en su tiempo libre. Según todos los modelos empresariales de SCA, no hay motivos para invertir menos en un servicio CSCA de calidad que en cualquier otro servicio paneuropeo. Si la Comisión Europea presta su apoyo a este servicio o, incluso, se encarga de su organización, habrá que cumplir una serie de obligaciones estrictas en cuanto a la calidad, el número de idiomas, la disponibilidad y la neutralidad del servicio, que a su vez deberá prestarse sobre la base de diversos acuerdos de nivel de servicio (*service level agreements, SLA*) con organismos profesionales.

Por lo tanto, hay que prever un presupuesto «normal» para este tipo de organización: una inversión inicial (prototipo) de entre 510 000 y 1 060 000 euros, los costes de implantación serán de 340 000 euros y el funcionamiento durante 5 años supondrá 975 000 euros anuales, lo que representa un coste total de propiedad (TCO) de aproximadamente 6 millones de euros en cinco años.

Los beneficios de la inversión no serán inmediatos, ya que, al principio, habrá que realizar un esfuerzo considerable para evaluar y «adaptar» el *software* existente para poderlo compartir como *software* de código abierto; por otra parte, los nuevos proyectos necesitarán tiempo hasta que lleguen a su madurez. Por eso, si se empieza de cero, consideramos que es necesario un compromiso de cuatro o cinco años para demostrar (tomando como base las estadísticas CSCA y las consultas de los usuarios) que los importes ahorrados son mayores que los invertidos y quiénes son los principales beneficiarios. Durante estos cinco primeros años será necesario financiar CSCA con fondos públicos. Es realista contar con que los beneficiarios iniciales no aportarán fondos: su principal función es proporcionar *software*. La industria informática de «equipos físicos y servicios» no tiene interés por este tipo de proyectos, ya que aspiran a concentrar sus esfuerzos en aplicaciones específicas (por ejemplo, sólo aplicaciones Linux para fomentar su mercado de servidores), que no son compatibles con la neutralidad de CSCA.