

Grado en Ingeniería Informática - Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática



FLOSS



Objetivo principal:
Conocer las nociones
básicas de FLOSS



Conceptos básicos

Breve historia

Modelos de gobernanza

Licencias

Ventajas de usar FLOSS

Investigación

Resumen

Conceptos básicos

- Imagina que tienes una app web que se llama “UVLHub” que sirve para crear datasets de distintas clases con distintas funcionalidades para consultar, subir, sincronizar, y gestionar en general datasets... La aplicación te la puedes bajar e instalar gratuitamente de internet



¿ Es FLOSS ?

Conceptos básicos



- El código no está disponible y por lo tanto no puede ser FLOSS
- Supongamos que el código está disponible en un repositorio accesible al público



Es ahora FLOSS



Conceptos básicos

- No se establece una licencia de uso y por lo tanto no es FLOSS
- Supongamos que el código está disponible y tiene asociada una licencia de uso



¿ Es ahora FLOSS ?

Conceptos básicos



¿ Es ahora FLOSS ?



Open
Source
License



¿Qué es FLOSS?

- *Un programa es software libre si los usuarios tienen las cuatro libertades esenciales: [\[1\]](#)*
 - *La libertad de **ejecutar** el programa como se desee, con cualquier propósito (libertad 0).*
 - *La libertad de **estudiar** cómo funciona el programa, y **cambiarlo** para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.*
 - *La libertad de **redistribuir** copias para ayudar a otros (libertad 2).*
 - *La libertad de **distribuir copias de sus versiones modificadas** a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.*

Extraído de <https://www.gnu.org/philosophy/free-sw.es.html>

¿Qué es NOT(FLOSS)?

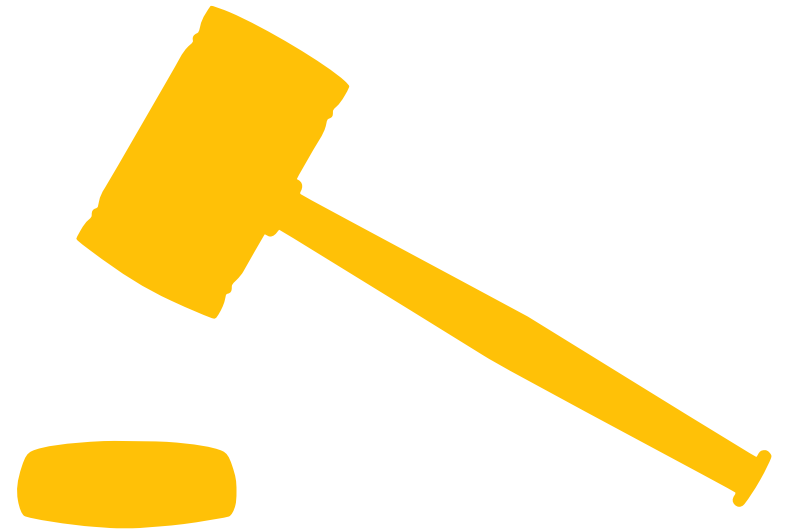
Los programas que no son FLOSS se conocen como “propietarios” o “software propietario” y son aquellos que limitan alguna de las 4 libertades del FLOSS

¿Existe alguna otra
posible
restricción?

¿Alguna restricción?

- ¡No existen restricciones!
- Ni tecnológicas
- Ni de campos concretos de actuación
- Ni de Hardware en el que se puede instalar

¿ Es ahora FLOSS ?



Discutamos para
buscar ejemplos de
qué sería y qué no
sería software libre



Conceptos básicos

Breve historia

Modelos de gobernanza

Licencias

Ventajas de usar FLOSS

Investigación

Resumen

Un poco de historia

- Al principio, todo el código era libre...
- En los años 50, todo el código se compartía por parte de investigadores/as, creadores/as, ingenieras¹ (la mayoría al principio eran ingenierAs y programadorAs de software, ¿lo sabías?) todo era “libre”



¹ https://elpais.com/tecnologia/2019/09/11/actualidad/1568211819_635943.html ,
https://www.eldiario.es/tecnologia/informatica-trabajo-mujeres-dominada-hombres_1_3226381.html

Un poco de historia

- En los años 60, apareció UNIX y forzó a vender el código por un lado y el hardware por otro

UNIX®
00011110 00011110 00011110 00011110 00011110 00011110 00011110 00011110



Un poco de historia

- En los años 70, aparecieron proyectos FLOSS importantes que todavía sobreviven



TEX

Un poco de historia

- En los años 80 empiezan las disputas más fuertes..
 - Problemas legales con UNIX BSD
 - Primera implementación libre de TCP/IP
 - Inicio del proyecto GNU: gcc,...
 - Aparecen otros proyectos: X11, ...
 - Se funda la FSF



free
SOFTWARE
FOUNDATION

Las grandes “guerras” en torno al software libre

- Unix vs SO libres
- Editores libres vs editores propietarios
- Windows vs Linux
- GNU vs KDE
- Docker vs Podman
- BSD vs GNU
- SL vs OSS

Un poco de historia

- En los años 90 aparecen ya un conjunto de proyectos con mucho peso
 - Linux (GNU/Linux)
 - Python
 - Debian
 - FreeBSD
 - Wine
 - PHP, GIMP, Ruby, Apache, KDE, GNOME, Netscape, OpenOffice
 - Se funda Red Hat

Un poco de historia

- En los años 2000 se consolidan los proyectos anteriores y aparecen nuevos
 - Blender
 - Firefox
 - Ubuntu
 - Git (2005)
 - Android (2007)
 - Chromium



Te quedan **5** artículos gratis este mes

SUSCRÍBETE

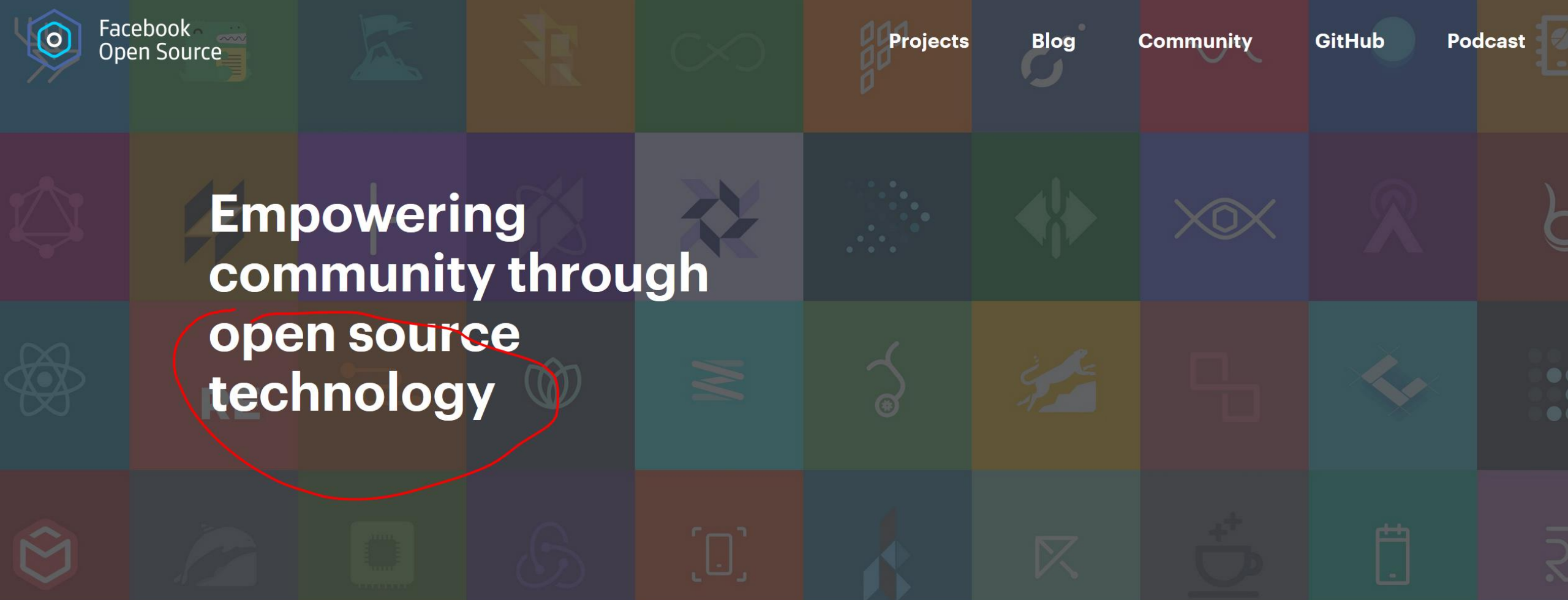
Microsoft admite que se equivocó con su guerra contra el 'software' libre

“La buena noticia es que, si la vida es lo suficientemente larga, puedes aprender... que necesitas cambiar”, asegura su presidente



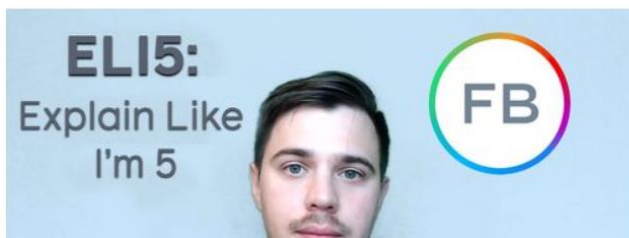
GUILLERMO VEGA

Las Palmas De Gran Canaria - 19 MAY 2020 - 19:30 CEST



Empowering community through open source technology

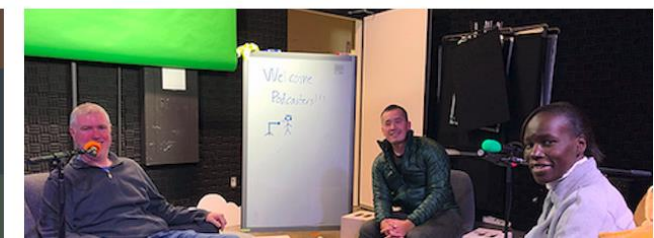
**Introducing ELI5:
Explain Like I'm 5 Series**



**Build a Smart Bookmarking Tool
with Rust and Rocket**



**The Diff: A Podcast From Facebook
Open Source**



Conceptos básicos

Breve historia



Modelos de gobernanza

Licencias

Ventajas de usar FLOSS

Investigación

Resumen

¿Cómo se os ocurre
que se puede
organizar todo esto?

Modelos de gobernanza de FLOSS

- **Corporativo:** una empresa controla el desarrollo, el código se libera, pero el proceso de desarrollo es cerrado. Ejemplo: Android, Red Hat Enterprise Linux,..
- **Dictador benevolente:** El proceso es abierto, pero las decisiones finales las toma una persona o muy pocas personas. Ejemplo: Kernel de Linux o Wikipedia
- **Modelos más democráticos:** hay más gente involucrada, hay más discusiones, más consenso, más mecanismos participativos,...Ejemplo: FreeBSD, Debian

Conceptos básicos

Breve historia

Modelos de gobernanza



Licencias

Ventajas de usar FLOSS

Investigación

Resumen

Licencias de software

- Definición: Contrato entre propietario de derechos de explotación y el usuario final
- La establece el propietario del software
- Una licencia gobierna:
 - Uso
 - Redistribución:
 - Código fuente
 - Binario ejecutable
 - Acceso al código fuente.
 - Monetización



Licencias de software

- Aceptación de licencia:
 - Descarga
 - Incorporación en producto
 - Scroll down + botón de “Aceptar”
- Licencia dual
- Relicenciamiento

Tipos de licencias

- Privativas (“propietarias”)
- No-copyleft
- Copyleft
- Dominio público



Licencias de software

- **Restrictivas:** BSD-like (promueve libertades individuales, no obliga al destinatario a volver a compartir el software)
- **Permisivas:** GPL-like (copyleft) (promueve libertades colectivas, obliga al destinatario de devolver mejoras a la comunidad)
- Mirar información aquí: <http://oss-watch.ac.uk/apps/licdiff/>
 - [Open Source Licenses by Category \(by Open Source Initiative\)](#): Lists open source licenses by category.
 - [Choose an Open Source License](#): Sponsored by GitHub, this site walks you through the properties you must consider, helping you decide what license makes sense.
 - [Browse Software Licenses & Summaries](#): Tags software licenses with key properties to help you navigate easier and better understand the existing licenses.
 - [Various Licenses and Comments About Them](#): Provides a description of copyleft licenses and comments about them.

Conceptos básicos

Breve historia

Modelos de gobernanza

Licencias



Ventajas de usar FLOSS

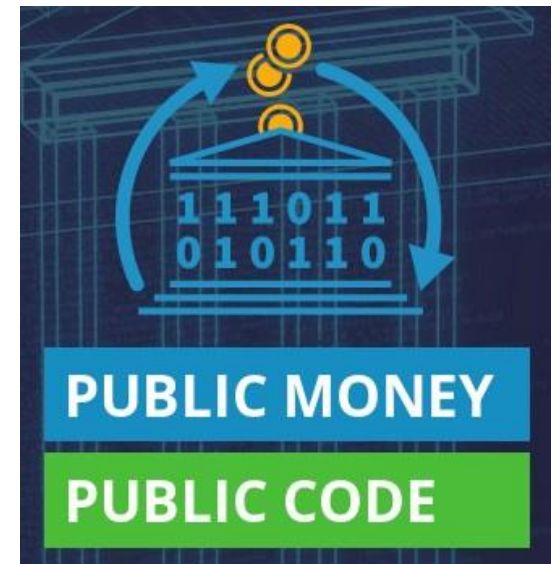
Investigación

Resumen

¿Existen ventajas e inconvenientes a la hora de usar FLOSS?

Ventajas de usar FLOSS

- Las ventajas de la transparencia: más gente viéndolo, reputación, calidad esperada,... (Revisión por pares)
- Promover la colaboración frente a la competencia
- Promover la reutilización y basarse en cosas hechas en lugar de reinventar la rueda
- Mayor independencia del proveedor y soporte técnico
- En administración pública, código como servicio público: “public money, public code”
- Acceso a la tecnología: estimula el aprendizaje
- Reduce la “obsolescencia programada”
- Estimula arreglar rápido los bugs
- ¿es más “verde”?
- ¿Ejemplos de proyectos FLOSS que conozcáis?



¿Y qué tiene que ver el
FLOSS con la
ingeniería del
software?

¿Y qué ha traído de bueno a la Ingeniería del Software el FLOSS?

- Buenas prácticas en automatización: git, CI, CD, *containers*, ...
- Lenguajes de programación
- Herramientas de desarrollo
- Buenas prácticas a la hora de desarrollar
- Reutilización de software, uno de los principios de IS
- ¡Muchos datos que analizar! *Mining Software Repositories*

Conceptos básicos

Breve historia

Modelos de gobernanza

Licencias

Ventajas de usar FLOSS



Investigación

Resumen

Trabajemos e investiguemos un rato

- Hacemos grupos de 3-5 personas
- Buscamos 5 proyectos de software
- Para cada uno ponemos:
 1. ¿Cuál fue el año de creación?
 2. ¿Qué licencia(s) de software tiene?
 3. ¿Qué lenguaje(s) de programación usa?
 4. ¿Qué repositorio de código usa?
 5. ¿Tiene gestión de *issues*?
 6. ¿Tiene integración continua?
 7. ¿Tiene pruebas automáticas?
 8. ¿Tiene análisis estático de código?
 9. ¿Tiene mecanismos de despliegue?
 10. ¿Qué modelo de gobernanza usa?

Conceptos básicos

Breve historia

Modelos de gobernanza

Licencias

Ventajas de usar FLOSS

Investigación

Resumen



Resumen

- El FLOSS es un concepto de plena actualidad
- En general se le presta poca atención a lo largo de la carrera
- Sin embargo, la industria tiene este concepto incorporado y se debe trabajar con él
- FLOSS no es sinónimo de gratis
- Deben respetarse las 4 libertades para que algo sea FLOSS
- Hay pugnas constantes dentro del mundo de FLOSS (licencias copyleft vs no copyleft)
- FLOSS ha aportado y aporta mucho tanto a la sociedad, a la educación y a la ingeniería de software

Algunas lecturas extra

Open Source Adoption Factors—A Systematic Literature Review

VÍCTOR REA SÁNCHEZ¹, PABLO NEIRA AYUSO², JOSÉ A. GALINDO², AND DAVID BENAVIDES¹

¹Facultad de Ciencias e Ingeniería, University of Málaga, Málaga 091050, Ecuador
²Departamento Lenguajes y Sistemas Informáticos, University of Seville, 41012 Seville, Spain

Corresponding author: Victor Rea Sánchez (vrcas@unemi.edu.ec)

This work was supported in part by the EU FEDER Program, in part by the MINECO Project OPHELIA under Grant RTI2018-101204-B-C22, in part by the Juan de la Cierva Postdoctoral Program, in part by the TASOVA Network under Grant MCIU-AEI TIN2017-90644-REDT, and in part by the Junta de Andalucía METAMORFOSIS Project.

ABSTRACT Nowadays, Free/Libre/OpenSource Software (FLOSS) is becoming a strategic option for many organizations in the public and the private sector. The lack of well defined guidelines for IT managers may jeopardize the FLOSS adoption process. FLOSS adoption procedures are developed ad-hoc in every organization, hence, leading to potential wheel reinvention situations. Identifying factors that influence and determine adoption is crucial. In this article, we survey existing literature through systematic review methodologies to make visible the technical, organizational and economic factors that must be evaluated in the adoption process. We also provide hints for researchers on publications and the type of research that already covered this topic in the past. We studied almost 500 papers from which we selected a final set of 54 primary studies directly related to FLOSS adoption. We found twenty-two different adoption factors categorized as technical (nine), organizational (nine) and economic (four). This article aims to provide the basic building blocks to step into the creation of a guide for the FLOSS adoption. All the data we used in this study is available at this online repository: <https://github.com/jagalindo/rea.victor.19-foss> and doi: <https://doi.org/10.5281/zenodo.2632543>

INDEX TERMS Free software adoption, open source adoption, libre software adoption.

I. INTRODUCTION

Free/Libre/OpenSource Software (FLOSS) is becoming ubiquitous. The ability to achieve a higher degree of vendor independence, interoperability and potential cost reductions are motivating organizations to consider FLOSS adoption [14], [18], [19]. A annual survey¹ highlights that 90 percent of respondents have already adopted FLOSS for their business in some way. According to this survey, adoptions are rapidly happening with no formal process and effective management. Problems in the adoption of FLOSS may set a bad precedent in the organization that may increase resistance for migrating to more FLOSS solutions among organization members, or even lead to a roll back to proprietary software solutions in some cases.

Identifying factors that influence the FLOSS adoption is a fundamental initial step to define guidelines. This allows IT managers and experts to evaluate potential risks and to

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Ardito.

¹10th Annual Future of Open Source Survey, <http://fbosp.northbridge.com/2016-future-open-source-survey-results>, including 1300 respondents from 64 countries

allocate the necessary resources to mitigate possible downsides. Factors such as compatibility with existing data formats in use in the organization, ease of use of the FLOSS software, availability of documentation, external support, maintenance and training among many others need to be conveniently evaluated. This problem is not new and it has already been covered, from different perspectives, in a significant amount of literature review contributions in the last two decades [1], [12], [21] and, according to the results that we provide in this article, we consider it an active domain of research. We define a set of factors that generally cover aspects relevant to a FLOSS evaluation. This article is intended to be an important basis for the construction of a guide for the adoption of FLOSS in public and private institutions.

This article revisits primary source literature through systematic review procedures to identify and to classify factors that literature highlights as relevant in the FLOSS adoption process. We followed a systematic review process inspired by the guidelines of [2] and [3] to extract relevant literature on FLOSS adoption. We filtered from more than 2,000 papers from the last twelve years to finally obtain 54 primary studies from which we gathered data and analyses. As a result,

<https://doi.org/10.1109/ACCESS.2020.2993248>



Brussels, 21.10.2020
C(2020) 7149 final

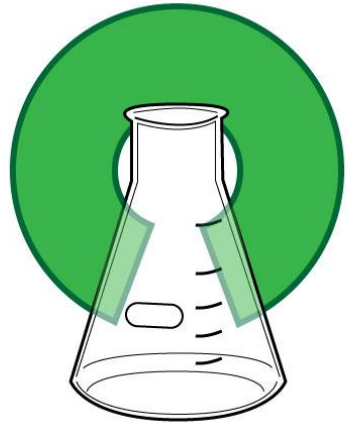
COMMUNICATION TO THE COMMISSION
OPEN SOURCE SOFTWARE STRATEGY 2020 – 2023
Think Open

EN

EN

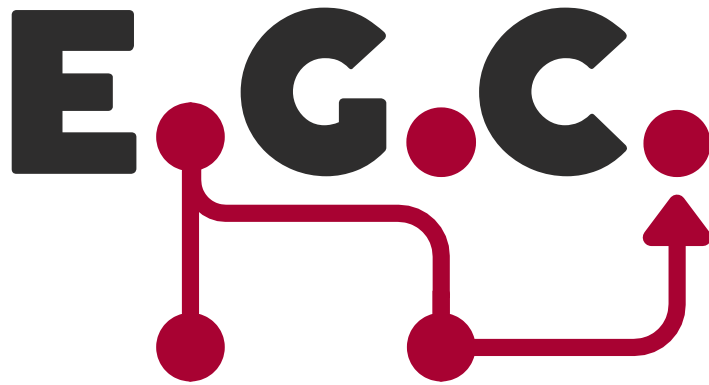
https://ec.europa.eu/info/sites/info/files/en_ec_open_source_strategy_2020-2023.pdf

Conceptos relacionados, pero no iguales



open science





Grado en Ingeniería Informática - Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática

¡Gracias!