



Código abierto Buen gobierno Manual

Autores: OSPO Alliance y los participantes en la GGI

Versión: v1.2

Fecha: 04-03-2024

Índice

1	Introducción	4
1.1	Contexto	4
1.2	Acerca de la Iniciativa de Buen Gobierno	5
1.3	Acerca de la OSPO Alliance	5
1.4	Traducciones	6
1.5	Contribuidores	6
1.6	Licencia	6
2	Organización	8
2.1	Terminología	8
2.2	Objetivos	8
2.3	Actividades canónicas	9
2.4	Cuadros de mando personalizados de actividades	10
3	InnerSource	11
3.1	¿Qué es InnerSource?	11
3.2	¿Por qué InnerSource?	11
3.3	La polémica del InnerSource	11
3.4	¿Quién lo hace?	12
3.5	InnerSource Commons, una referencia esencial	12
3.6	Diferencias en la gestión de InnerSource	12
4	Metodología	14
4.1	Preparar el terreno	14
4.2	Flujo de trabajo	14
4.3	Configuración manual: usar cuadros de mando personalizados de actividades	15
4.4	Configuración automática: usar la funcionalidad de despliegue de GGI	16
4.5	Disfrute	17
5	Actividades del objetivo de uso	18
5.1	Inventario de competencias y recursos de código abierto	18
5.2	Aumento de la competencia de código abierto	19
5.3	Supervisión del código abierto	21
5.4	Software empresarial de código abierto	22
5.5	Gestionar competencias y recursos de código abierto	24
6	Actividades del objetivo de confianza	26
6.1	Gestionar el cumplimiento de la legislación	26
6.2	Gestionar las vulnerabilidades del software	28
6.3	Gestionar las dependencias del software	30
6.4	Gestionar los indicadores clave	31
6.5	Revisión del código	33
7	Actividades del objetivo de cultura	35
7.1	Promover las mejores prácticas de desarrollo del código abierto	35
7.2	Contribuir a los proyectos de código abierto	36
7.3	Pertenecer a la comunidad de código abierto	38
7.4	Perspectiva de los recursos humanos	39
7.5	Cambio en origen	41
8	Actividades del objetivo de compromiso	43
8.1	Participar en proyectos de código abierto	43
8.2	Apoyar a las comunidades de código abierto	44
8.3	Hacer público el uso del código abierto	45
8.4	Comprometerse con los proveedores de código abierto	46
8.5	Política de adquisición de código abierto	47
9	Actividades del objetivo de estrategia	49
9.1	Establecer una estrategia para la gobernanza corporativa del código abierto	49
9.2	Conciencia de la gerencia	50

9.3 Fuente abierta y soberanía digital	51
9.4 El código abierto hace posible la innovación	53
9.5 El código abierto al servicio de la transformación digital	54
10 Conclusión	57
10.1 Contacte con	57
10.2 Apéndice: Plantilla de cuadro de mando de actividades personalizado	57

1 Introducción

Este documento presenta una metodología para implantar una gestión profesional del software de código abierto en una organización. Aborda la necesidad de utilizar software de código abierto de forma adecuada y justa, salvaguardar a empresa de amenazas técnicas, legales y de propiedad intelectual y maximizar las ventajas del software de código abierto. Donde sea que se encuentre la organización respecto a estos temas, este documento propone orientación e ideas para avanzar y hacer de su viaje un éxito.

1.1 Contexto

La mayoría de los grandes usuarios e integradores de sistemas ya utilizan Software Libre y de Código Abierto (FOSS) ya sea en sus sistemas de información o en sus divisiones de productos y servicios. El cumplimiento de las condiciones legales del código abierto viene siendo una preocupación creciente, y muchas empresas grandes han nombrado responsables de cumplimiento. Sin embargo aunque asegurar su cadena de producción de software código abierto - que es la esencia del cumplimiento - es fundamental, los usuarios *tienen* que contribuir a las comunidades para la sostenibilidad del ecosistema de código abierto. Vemos a la gobernanza de código abierto acompañando a todo el ecosistema, participando en las comunidades locales, nutriendo una relación sana con los proveedores de software de código abierto y sus servicios asociados. Esto eleva a la conformidad al siguiente nivel y es de lo que trata un "buen" gobierno del código abierto.

Esta iniciativa abarca más allá del cumplimiento y la responsabilidad legal. Trata acerca de desarrollar conciencia en las comunidades de usuarios finales (muchas veces desarrolladores de software a su vez) e integradores de sistemas. Y de desarrollar relaciones mutuamente beneficiosas en el seno del ecosistema europeo de FOSS.

La Buena Gobernanza del OSS habilita a organizaciones de todo tipo -- empresas pequeñas y grandes, ayuntamientos, universidades, asociaciones, etc. -- para maximizar los beneficios derivados del código abierto ayudo a alinear personas, procesos, tecnología y estrategia. Y en esta área, la de maximizar las ventajas del código abierto, especialmente en Europa, todo el mundo está aún aprendiendo e innovando, y nadie sabe donde se encuentra realmente respecto al estado del arte.

Esta iniciativa intenta ayudar a las organizaciones a lograr estos objetivos mediante:

- Un catálogo estructurado de **actividades**, un plan rector para la implantación de una gestión profesional del software de código abierto.
- Una **herramienta de gestión** para definir, monitorizar, informar y comunicar el progreso.
- Un **camino claro y práctico para la mejora**, con pasos pequeños y accesibles para mitigar riesgos, educar a las personas, adaptar procesos y comunicar interna y externamente al dominio de la organización.
- **Orientación** y una serie de **referencias seleccionadas** acerca del licenciamiento de código abierto, prácticas recomendadas, formación, y participación en el ecosistema para elevar la concienciación y la cultura del código abierto, consolidar el conocimiento interno y extender el liderazgo.

Este guía se ha desarrollado contemplando los siguientes requisitos:

- Se cubre cualquier tipo de organización: desde PYMEs a grandes empresas y ONGs, desde autoridades locales (ayuntamientos) a grandes instituciones (europeas o gubernamentales). El marco de trabajo proporciona piezas para componer una estrategia y pistas para a su realización, pero *cómo* se ejecuten las actividades depende totalmente del contexto propio del programa y se deja a criterio de su gestor. Podría ser útil buscar servicios de consultoría e intercambiar ideas con otros pares.
- No se da nada por sentado acerca del nivel de conocimiento técnico en la organización o el dominio de actividad. Por ejemplo, algunas organizaciones necesitarán establecer un currículo formativo completo mientras que otras quizá simplemente propongan material adaptado a los equipos.

Algunas actividades no serán relevantes en todas las situaciones pero el marco de trabajo en sí sigue proporcionando una hoja de ruta completa y allana el camino a estrategias personalizadas.

1.2 Acerca de la Iniciativa de Buen Gobierno

En OW2, una iniciativa es un esfuerzo conjunto para abordar una necesidad de mercado. La [Iniciativa por el Buen Gobierno](#) propone un marco metodológico para implantar en las organizaciones una gestión profesional del software de fuentes abiertas.

La Iniciativa de Buen Gobierno se basa en un modelo completo inspirado en la conocida jerarquía de necesidades y motivaciones humanas de Abraham Maslow; tal como ilustra la siguiente imagen.



Jerarquía de motivos de conducta de Abraham Maslow

Objetivos de OW2 para la buena gobernanza del OSS

Mediante ideas, directrices y actividades, la Iniciativa de Buen Gobierno proporciona un patrón para la implantación de entidades organizativas comisionadas para la gestión profesional del software de fuentes abiertas, también llamadas OSPO (por las siglas en inglés de Oficina de Programa de Software de Fuentes Abiertas). La metodología es también un sistema de gestión para definir prioridades y monitorizar y compartir avances.

A medida que las organizaciones implantan la metodología de Buen Gobierno del OSS ampliarán sus habilidades en varias direcciones, incluyendo:

- **usar** software de fuentes abiertas de forma adecuada y segura en la empresa para mejorar la reutilización y mantenibilidad y la velocidad del desarrollo del software;
- **mitigar** los riesgos legales y técnicos asociados al código externo y a la colaboración;
- **identificar** la formación necesaria para los equipos, desde los programadores a los jefes de equipo y gestores, para que todos compartan la misma visión;
- **priorizar** objetivos y actividades, para desarrollar una estrategia eficiente de código abierto;
- **comunicar** eficientemente en la empresa y al mundo exterior para aprovechar la estrategia de código abierto;
- **mejorar** la competitividad y hacer la organización atractiva para el talento estrella del código abierto.

1.3 Acerca de la OSPO Alliance

Las principales organizaciones europeas sin ánimo de lucro del código abierto, incluyendo a OW2, la Eclipse Foundation, el OpenForum Europe y la Foundation for Public Code, lanzaron la **OSPO Alliance** con la misión de desarrollar conciencia por el código abierto en Europa y en el mundo y de promover una gestión estructurada y profesional del código abierto en empresas y administraciones.

Mientras la iniciativa por el Buen Gobierno se enfoca en desarrollar una metodología de gestión, la OSPO Alliance tiene el objetivo más amplio de ayudar a empresas, particularmente en los sectores no tecnológicos, e instituciones públicas, a descubrir y comprender el código abierto, comenzar a beneficiarse de él en sus actividades y crecer para auspiciar sus propias OSPOs.

A OSPO Alliance ha establecido el sitio web **OSPO.Alliance** alojado en <https://ospo-alliance.org>. La OSPO Alliance sirve a la comunidad con un lugar seguro en el que debatir e intercambiar ideas acerca de temas de OSPOs y proporciona un repositorio para un conjunto completo de recursos para corporaciones, instituciones públicas y organizaciones académicas y de

investigación. La OSPO Alliance conecta con OSPOs en toda Europa y en el mundo, así como con organizaciones comunitarias que la apoyan. Promueve buenas prácticas y fomenta la contribución a la sustentabilidad del ecosistema de código abierto. Visite el sitio web [OSPO Alliance](#) para una rápida visión general de las estructuras complementarias de buenas prácticas de gestión de TI.

El sitio web [OSPO Alliance](#) es también el lugar donde recabamos comentarios de la comunidad sobre la iniciativa y sus contenidos (actividades, acervo de conocimiento).

1.4 Traducciones

Este libro se escribió originalmente en inglés. También está disponible en francés, alemán, portugués, holandés, italiano y español, gracias a un trabajo comunitario en curso para traducir el Manual de la GGI. Dado que los avances evolucionan rápidamente, le recomendamos que consulte nuestra página web oficial para obtener una lista completa de las traducciones disponibles.

Vea <https://ospo-alliance.org/ggi/>

El manual de GGI se traduce empelando [Weblate](#), un proyecto y plataforma de código abierto que ofrece hospedaje gratuito para proyectos de código abierto. Queremos agradecerles profundamente, así como a todos nuestros colaboradores de traducción. Sois alucinantes.

Vea <https://hosted.weblate.org/projects/ospo-zone-ggi/#languages>

1.5 Contribuidores

Las siguientes grandes personas han contribuido al manual de la Iniciativa para la Buena Gobernanza:

- Frédéric Aatz (Microsoft Francia)
- Boris Baldassari (Castalia Solutions, Eclipse Foundation)
- Philippe Bareille (Ayuntamiento de París)
- Gaël Blondelle (Eclipse Foundation)
- Vicky Brasseur (Wipro)
- Philippe Carré (Nokia)
- Pierre-Yves Gibello (OW2)
- Michael Jaeger (Siemens)
- Sébastien Lejeune (Thales)
- Max Mehl (Free Software Foundation Europe)
- Catherine Nuel (OW2)
- Hervé Pacault (Orange)
- Stefano Pampaloni (RIOS)
- Christian Paterson (OpenUp)
- Simon Phipps (Meshed Insights)
- Silvério Santos (Orange Business)
- Cédric Thomas (OW2)
- Nicolas Toussaint (Orange Business)
- Florent Zara (Eclipse Foundation)
- Igor Zubiaurre (Bitergia)

1.6 Licencia

Esta obra está licenciada mediante licencia [Creative Commons Attribution 4.0 International](#) (CC-BY 4.0). Del sitio web de Creative Commons:

Tiene Ud. derecho a:

- Compartir — copiar y redistribuir el material en cualquier soporte o formato
- Adaptar — remezclar, transformar, y crear partiendo del material

para cualquier propósito, incluso comercialmente.

Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de

forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. [Original: <https://creativecommons.org/licenses/by/4.0/deed.es>].

Todo el contenido es Copyright de OSPO Alliance y otros.

2 Organización

2.1 Terminología

El diseño de la metodología de Buen Gobierno de OSS está estructurado en torno a cuatro conceptos clave: objetivos, actividades canónicas, cuadros de mando de actividades personalizadas e iteración.

- **Objetivos:** un objetivo es un conjunto de actividades asociadas a un área de interés común. Existen cinco objetivos: objetivo de utilización, objetivo de confianza, objetivo de cultura, objetivo de desarrollo y objetivo de estrategia. Los objetivos se pueden alcanzar de forma independiente, en paralelo, o refinándolos iterativamente por actividades.
- **Actividades canónicas:** dentro de un objetivo, cada actividad aborda un único asunto de desarrollo - como la gestión de cumplimiento legal - que puede aprovecharse como un paso incremental en dirección a los objetivos del programa. El conjunto completo de actividades que define la GGI se denomina Actividades Canónicas.
- **Cuadro de mando personalizado actividades (CAS):** para implementar GGI en una determinada organización, las actividades canónicas se deben adaptar a las especificidades del contexto, constituyendo así un conjunto de cuadros de mando personalizados de actividades. El cuadro de mando personalizado de actividades describe como se implementará la actividad en el contexto de una organización y como se monitorizará su progreso.
- **Iteración:** El Buen Gobierno de OSS es un sistema de gestión y como tal requiere evaluaciones y revisiones periódicas. Piense en el sistema de contabilidad de una organización, es un proceso continuo con al menos un punto de control anual, el balance; de la misma manera, el proceso de Buen Gobierno de OSS requiere al menos una revisión anual, no obstante, puede haber revisiones parciales o más frecuentes, dependiendo de las actividades.

2.2 Objetivos

As actividades canónicas que define GGI se organizan en objetivos. Cada objetivo aborda un área específica de avance en el proceso. Desde la utilización a la estrategia, los objetivos cubren cuestiones relacionadas con todos los afectados, desde los equipos de desarrollo al nivel ejecutivo.

- **Objetivo Uso:** este objetivo cubre los pasos básicos en la utilización de software de código abierto. Las actividades relacionadas con el objetivo de utilización cubren los primeros pasos de un programa de código abierto, identificando la eficiencia de uso del código abierto y qué aporta éste a la organización. Incluye formación y gestión de conocimiento, produciendo inventarios de software de código abierto en uso interno, y presenta conceptos de código abierto que se pueden utilizar a lo largo de todo el proceso.
- **Objetivo Confianza:** este objetivo trata el uso seguro del código abierto. El objetivo de confianza trata el cumplimiento legal, la gestión de dependencias y vulnerabilidades, y trata de generar confianza respecto a la forma en que la organización emplea el código abierto.
- **Objetivo Cultura:** el objetivo de cultura incluye actividades orientadas a hacer que los equipos se sientan cómodos con el código abierto, participando individualmente en actividades colaborativas, comprendiendo e implementando buenas prácticas de código abierto. Este objetivo fomenta entre los individuos un sentimiento de pertenencia a la comunidad del código abierto.
- **Objetivo Involucración:** este objetivo busca involucrarse en el ecosistema del código abierto a nivel corporativo. Los recursos humanos y financieros están presupuestados para contribuir a proyectos de código abierto. Aquí, la organización afirma ser un ciudadano responsable del código abierto y reconoce su responsabilidad en asegurar la sostenibilidad del ecosistema de código abierto.
- **Objetivo Estrategia:** este objetivo trata de hacer al código abierto visible y aceptable en los niveles más altos de la gerencia corporativa. Se trata de reconocer que el código abierto es un facilitador estratégico de la soberanía digital, proceso de innovación y, en general, una fuente de atracción y buena voluntad.

2.3 Actividades canónicas

Las actividades canónicas son centrales en el diseño de GGI. En su versión inicial, la Metodología GGI ofrece cinco actividades canónicas por objetivo, 25 en total. Las actividades canónicas se describen empleando las siguientes secciones predefinidas:

- *Descripción*: un resumen del asunto que la actividad aborda y de los pasos para completarla.
- *Evaluación de oportunidades*: describe cuando y por qué es importante realizar esta actividad.
- *Evaluación de progreso*: describe como medir el avance de la actividad y evaluar su éxito.
- *Herramientas*: una lista de tecnologías y/o herramientas que pueden ayudar a realizar esta actividad.
- *Recomendaciones*: sugerencias y buenas prácticas coleccionadas por los participantes en GGI.
- *Recursos*: enlaces y referencias para leer más sobre el tema tratado en la actividad.

Descripción

Esta sección aporta una descripción de alto nivel de la actividad, un resumen del asunto para establecer el propósito de la actividad en contexto con el enfoque de código abierto en el marco de un objetivo.

Evaluación de oportunidades

Para ayudar a estructurar un enfoque iterativo, cada actividad tiene una sección "Evaluación de oportunidades" con una o más cuestiones relacionadas. La evaluación de oportunidades se centra en por qué es importante realizar esta actividad; qué necesidades aborda. Evaluar la oportunidad ayudará a estimar esfuerzos, recursos necesarios, y ayudará a evaluar e los costes y el retorno esperado de la inversión (ROI).

Evaluación del progreso

Este paso se centra en definir los objetivos, KPI (Indicadores Clave de Rendimiento¹), y en proporcionar *puntos de verificación* que ayuden a evaluar el progreso en la Actividad. Los puntos de verificación que se sugieren, pueden ayudar a definir una hoja de ruta para el proceso de Buena Gobernanza, sus prioridades y cómo se medirán los progresos.

Herramientas

Aquí se listan las herramientas que pueden ayudar a entregar la actividad o instrumentar un paso específico de las actividades. Las herramientas no son obligatorias ni pretenden ser exhaustivas, sino que son sugerencias o categorías sobre las que trabajar en base a un contexto existente.

Recomendaciones

Esta sección se actualiza regularmente con comentarios de los usuarios y todo tipo de recomendaciones que puedan ayudar a gestionar la actividad.

Recursos

Los recursos se proponen para alimentar el enfoque con estudios de fondo, documentos de referencia, eventos o contenido on-line para enriquecer y desarrollar el enfoque relacionado con la actividad. Los recursos no son exhaustivos, sino puntos de partida o sugerencias para expandir la semántica de la actividad según su propio contexto.

¹Un indicador de rendimiento o indicador clave de rendimiento es un tipo de medida del rendimiento. Los KPI evalúan el progreso y el éxito de una organización o de una actividad concreta a la que se dedique.

2.4 Cuadros de mando personalizados de actividades

Los cuadros de mando personalizados de actividades (CAS) son ligeramente más detallados que las actividades canónicas. Un CAS incluye detalles específicos de la organización que implementa GGI. El uso de CAS se describe en la sección Metodología.

3 InnerSource

InnerSource es cada vez más popular en las empresas, ya que ofrece un enfoque basado en prácticas de código abierto que han tenido éxito en los equipos de desarrollo de las organizaciones. Sin embargo, hacer InnerSource no es simplemente copiar y pegar esas prácticas. Hay que adaptarlas a la cultura y la organización interna propias de cada empresa. Veamos con más detalle qué es y qué no es InnerSource, y cuáles son los retos asociados.

3.1 ¿Qué es InnerSource?

El término fue utilizado por primera vez por Tim O'Reilly en 2000, al afirmar que Innersourcing es "[...] *el uso de técnicas de desarrollo de código abierto dentro de la empresa.*"

Según [InnerSource Commons](#), la fundación de referencia sobre el tema, InnerSource es "*uso de principios y prácticas de código abierto para el desarrollo de software dentro de los confines de una organización.*"

3.2 ¿Por qué InnerSource?

Sin embargo, según [InnerSource Commons](#), "*para las empresas que construyen principalmente software de código cerrado, InnerSource puede ser una gran herramienta para ayudar a romper los silos, fomentar y ampliar la colaboración interna, acelerar la incorporación de nuevos ingenieros e identificar oportunidades para contribuir con software al mundo del código abierto*"

Es interesante observar que las ventajas de InnerSource pueden repercutir en diversas funciones de una empresa, no sólo en la ingeniería. Como resultado, algunas empresas han encontrado ventajas concretas en áreas como:

- Funciones jurídicas: acelerar el establecimiento de colaboraciones interfuncionales mediante el uso de un marco jurídico listo para usar (licencia InnerSource).
- Recursos humanos: gestión de competencias escasas a través de un equipo central experimentado responsable de aunar esfuerzos y conocimientos.

3.3 La polémica del InnerSource

InnerSource está rodeado de mitos que uno puede oír de sus detractores. Aunque no es verdadero código abierto, muestra grandes beneficios potenciales para las organizaciones que despliegan este enfoque internamente. He aquí algunos de estos mitos:

- [MITO] InnerSource se hace a expensas del código abierto (principalmente saliente):
 - Los proyectos de software permanecen detrás del cortafuegos de la empresa.
 - Menos contribuciones externas al código abierto.
- [MITO] Secuestrar el espíritu del código abierto frente a acercarse a él.
- [MITO] Ningún proyecto de InnerSource se ha convertido nunca en un proyecto de código abierto.
- [MITO] El motivo para hacer InnerSource es que es similar al código abierto. Pero en realidad, si un desarrollador lo valora, siempre debería preferirse una contribución directa de código abierto.

He aquí algunos datos sobre la práctica de InnerSource que desmienten la mayoría de los mitos anteriores:

- [HECHO] InnerSource es una forma de dar la bienvenida al código abierto a empresas principalmente cerradas.
- [HECHO] Aunque la mayoría de las contribuciones al código abierto las realizan voluntarios, podemos publicitar la participación en el código abierto a los ingenieros utilizando esta lista de "beneficios percibidos".
- [HECHO] En algunos casos (¿o en la mayoría?), las empresas no siguen una práctica de desarrollo ordenada y controlada, y esto (GGI) puede ser una forma de ayudarles a gestionarlo.
- [HECHO] Aún será necesario * mucho * de trabajo en la conversión de licencias cerradas a abiertas.

- [HECHO] Efectivamente, hay casos de proyectos InnerSource de código abierto:
 - Twitter Bootstrap.
 - Kubernetes de Google.
 - Docker de dotCloud (nombre anterior de Docker Inc.).
 - React Native.
- [HECHO] El código abierto se beneficia del aumento de ingenieros de software que se familiarizan con las prácticas del código abierto, ya que las de InnerSource son muy similares.

3.4 ¿Quién lo hace?

Muchas empresas han puesto en marcha iniciativas InnerSource o ISPO (InnerSource Program Office), algunas desde hace tiempo, otras más recientemente. He aquí una lista no exhaustiva, centrada principalmente en empresas europeas:

- Banco de Santander ([fuente](#))
- BBC ([fuente](#))
- Bosch ([fuente](#))
- Comcast ([fuente](#))
- Ericsson ([fuente](#))
- Engie ([fuente](#))
- IBM ([fuente](#))
- Mercedes ([fuente](#))
- Microsoft ([fuente](#))
- Nike ([fuente](#))
- Nokia ([fuente](#))
- SNCF Connect & Tech ([fuente 1](#), [fuente 2](#))
- Paypal ([fuente](#))
- Philips([fuente](#))
- Renault ([fuente](#))
- SAP ([fuente](#))
- Siemens([fuente](#))
- Société Générale([fuente](#))
- Thales ([fuente](#))
- VeePee([fuente](#))

3.5 InnerSource Commons, una referencia esencial

En [InnerSource Commons](#) se puede encontrar una comunidad activa y vibrante de practicantes de InnerSource, que trabaja según los principios del código abierto. Proporcionan un montón de recursos útiles para ponerse al día en la materia, incluyendo [patrones](#), una [ruta de aprendizaje](#) y pequeños libros electrónicos:

- [Comenzando con InnerSource](#) por Andy Oram.
- [Comprensión de la lista de verificación de InnerSource](#) por Silona Bonewald.

3.6 Diferencias en la gestión de InnerSource

InnerSource trae desafíos específicos que no se enfrentan en Open Source. Sin embargo, la mayoría de las organizaciones que crean software privativo ya se están ocupando de ellos:

- Licencia dedicada y específica de la empresa para proyectos Innersource (para grandes empresas con múltiples entidades jurídicas).
- La naturaleza pública del código abierto lo salva de los desafíos de los precios de transferencia. La naturaleza privada de InnerSource expone a las empresas que operan en diferentes jurisdicciones a responsabilidades por transferencia de beneficios.
- Las motivaciones para contribuir son muy diferentes:
 - InnerSource tiene un grupo más reducido de posibles contribuyentes porque se limita a la organización.
 - Mostrar las propias habilidades profesionales es un motor para contribuir. InnerSource limita este impacto únicamente a los límites de la organización.

- Contribuir a mejorar la sociedad es otro de los motores de las contribuciones limitadas en InnerSource.
- Por lo tanto, la motivación requiere un mayor esfuerzo y se basará más en recompensas y tareas.
- Afrontar los miedos al perfeccionismo, como el síndrome del impostor, es más fácil en InnerSource debido a la visibilidad limitada del código.
- La externalización de la mano de obra es más frecuente, lo que afecta a la gobernanza de varias maneras.
- Evaluar la adecuación de la empresa es más fácil para InnerSource porque se desarrolla internamente.
- La facilidad de búsqueda tiende a convertirse en un problema. Indexar la información es menos prioritario para las empresas. Los motores de búsqueda públicos como DuckDuckGo, Google o Bing hacen un trabajo mucho mejor que InnerSource no puede aprovechar.
- InnerSource está en una posición ligeramente mejor para controlar la exportación, ya que vive dentro de la empresa.
- Se necesita un control de los límites de la IP que se filtra como código fuente.

InnerSource sigue evolucionando a medida que más empresas adoptan sus principios y comparten sus experiencias. En una versión posterior de este manual se ofrecerá una lista de las actividades del GGI relevantes para los profesionales de InnerSource.

4 Metodología

Implementar la metodología de Buen Gobierno de OSS es a la postre una iniciativa con consecuencias e impacto. Implica varias categorías de personas, servicios y procesos de empresa, desde las prácticas cotidianas hasta la gestión de RRHH y desde los desarrolladores hasta el nivel ejecutivo. No hay realmente un mecanismo mágico para implementar el buen gobierno del código abierto. Diferentes tipos de organización, y culturas y circunstancias empresariales requerirán enfoques diferentes del buen gobierno del código abierto. Para cada organización, habrá limitaciones e expectativas distintas, llevando a diferentes rutas y maneras de gestionar el programa.

Con esto en mente, la Iniciativa para el Buen Gobierno proporciona un diseño genérico de actividades que se pueden adaptar al dominio, cultura y requisitos de propios de la organización. Si bien el diseño pretende ser completo la metodología se puede implementar progresivamente. Es posible iniciar el programa simplemente seleccionando los objetivos y actividades más relevantes en cada contexto específico. La idea es construir un primer borrador del libro de ruta para ayudar a establecer la iniciativa local.

Junto con este marco de trabajo también recomendamos encarecidamente contactar con colegas a través de una red bien establecida como la europea [OSPO Alliance](#), u otras iniciativas semejantes del TODO group, o de OSPO++. Lo que importa es poder intercambiar ideas con personas que lleven una iniciativa similar y compartir los problemas que surgen y las soluciones existentes.

4.1 Preparar el terreno

Dada a ambición de la metodología GGI y su amplio impacto potencial es importante comunicarse con una variedad de personas dentro de una organización. Sería apropiado involucrarlas para establecer un conjunto inicial de expectativas y requisitos realistas para comenzar con buen pie, atraer interés y apoyos. Una buena idea podría ser publicar los cuadros de mando personalizados de actividades en la plataforma colaborativa de la organización para que se puedan utilizar para comunicarse con los afectados. Algunos consejos:

- Identifique a los principales afectados, que acuerden un conjunto de objetivos primarios. Involúcrelos en el éxito de la iniciativa como parte de sus propias agendas.
- Obtenga un compromiso inicial, acuerde las etapas y el ritmo y establezca controles regulares para informarles de los avances.
- Asegúrese de que entiendan los beneficios de lo que se puede lograr y lo que implica: la mejora esperada debe ser clara y el resultado visible.
- Establezca un primer diagnóstico del estado de arte del código abierto en la organización candidata. Resultado: un documento describiendo lo que este programa logrará, donde está la organización y a dónde pretende ir.

4.2 Flujo de trabajo

Como profesionales de software modernos, nos gustan los métodos agilistas que definen incrementos pequeños y seguros, dado que es una buena práctica reevaluar la situación regularmente y proporcionar unos mínimos resultados intermedios significativos.

En el contexto de un programa vivo de OSPO esto es muy importante ya que muchos aspectos secundarios evolucionarán, desde la estrategia y respuesta de la organización al código abierto hasta la disponibilidad y compromiso de las personas. La reevaluación e iteración periódicas también permiten adaptarse a la aceptación del programa en curso, seguir mejor las tendencias y oportunidades actuales, y obtener beneficios incrementales para los afectados y para la organización en su totalidad.

Idealmente la metodología podría implementarse en cinco fases como se indica:

1. **Descubrimiento** Entender los conceptos-clave, asimilar la metodología, alinear las metas y las expectativas.
2. **Adecuación** Adaptación de la descripción de la actividad y la evaluación de oportunidades a las especificidades de la organización.

3. **Priorización** Identificación de objetivos y resultados clave, tareas y herramientas, planificación de hitos y esbozo del cronograma.
4. **Activación** Finalización del cuadro de mando, presupuesto, asignaciones, documentación en el gestor de tareas.
5. **Iteración** Evaluar y puntuar resultados, destacar asuntos, mejorar, ajustar. Iterar todos los trimestres u semestres.

Preparar la primera iteración del programa:

- Identifique un primer conjunto de tareas sobre las que trabajar, y priorícelas de acuerdo a las necesidades (brechas hasta el estado deseado) y el cronograma. Resultado: una lista de tareas sobre las que trabajar, durante la iteración.
- Defina un conjunto de requisitos y áreas de mejora, comuníquelo a los afectados y usuarios finales, obtenga su aprobación o compromiso.
- Rellene los cuadros de mando para trazar el progreso. Se puede descargar una plantilla del cuadro de mando de [repositório GGI](#).

Al final de cada iteración, haga una retrospectiva y prepárese para a próxima iteración:

- Comunique las últimas mejoras.
- Evalúe donde se encuentra, si se han concluido las tareas previstas, refine la hoja de ruta en consecuencia.
- Compruebe los asuntos pendientes, si es necesario solicite apoyo a otros actores o servicios.
- Redefina las tareas de acuerdo al contexto actualizado.
- Defina un nuevo subconjunto de tareas a ejecutar.

4.3 Configuración manual: usar cuadros de mando personalizados de actividades

Un cuadro de mando personalizado de actividades es una forma que describir una actividad canónica adaptada a las especificidades de una organización. En conjunto, los cuadros de mando personalizados de actividades proporcionan una hoja de ruta para a gestión del software de código abierto.

Observe que según la experiencia inicial con la metodología, adaptar una actividad canónica a un cuadro de mando específico para una organización puede llevar hasta una hora .

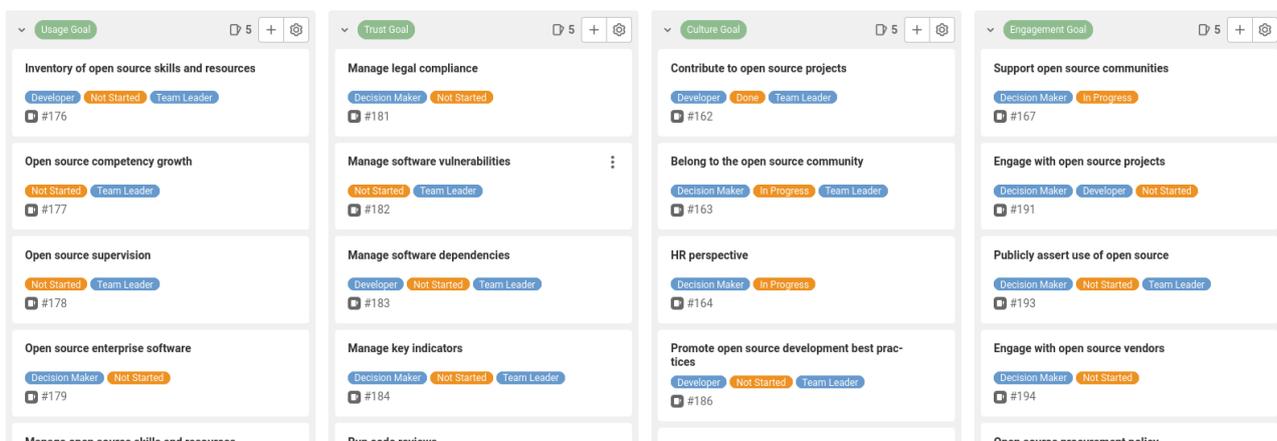
Un cuadro de mando personalizado de actividades contiene las siguientes secciones:

- **Desambiguación del título** Antes de nada, dedique algunos minutos a entender de qué va la actividad y su relevancia, como puede encajar en su gestión general del OSS.
- **Descripción personalizada** Adapte la actividad a las especificidades de la organización. Defina el ámbito de la actividad, el caso de uso particular que va a abordar.
- **Evaluación de oportunidades** Explique porque es importante emprender esta actividad, qué necesidades aborda. ¿Cuales son los puntos de dolor? ¿Cuales son las oportunidades para avanzar? ¿Qué se puede ganar?
- **Objetivos** Defina un par de objetivos cruciales para a actividad. Puntos de dolor que arreglar, oportunidades de avance, deseos. Identifique tareas clave. Lo que intentamos lograr en esta iteración.
- **Herramientas** Tecnología, herramientas y productos empleados en la actividad.
- **Notas operacionales** Indicaciones sobre el enfoque, el método, la estrategia para avanzar en esta actividad.
- **Resultados clave** Defina resultados esperados medibles y verificables. Escoja resultados que indiquen el avance respecto a los objetivos. Indique aquí los KPIs.
- **Progreso y puntuación** Progreso es el grado de completitud del resultado, en %; Puntuación es la valoración personal del éxito.
- **Evaluación personal** Puede Ud. añadir una breve explicación para cada resultado y para el grado de satisfacción personal expresado en su puntuación.
- **Cronograma** Indique fechas de inicio y fin, fases, pasos críticos, hitos.
- **Esfuerzos** Evalúe el tiempo y los recursos materiales solicitados, internos y de terceros. ¿Qué esfuerzos se esperan? ¿Cuanto va a costar? ¿Qué recursos necesitamos?
- **Responsables** Diga quién participa. Asigne tareas o liderazgo y responsabilidades de la actividad.

- **Obstáculos** Identifique los retos principales, dificultades previstas, riesgos, bloqueos, incertidumbres, puntos de atención, dependencias críticas.
- **Estado** Escriba aquí una evaluación sucinta de cómo progresa la actividad: ¿adecuadamente? ¿atrasada? Etc.
- **Evaluación global de progreso** Su propia evaluación de progreso de la actividad a alto nivel, sucinta, orientada a la gestión.

4.4 Configuración automática: usar la funcionalidad de despliegue de GGI

A partir de la versión 1.1 del Manual, la GGI propone **My GGI Board**, una herramienta automatizada para desplegar su propia instancia de GGI en la forma de un proyecto en GitLab. Configurar el proceso de instalación lleva menos de 10 minutos, está totalmente documentado y proporciona una manera simple y fiable de personalizar las actividades, acompañar su ejecución a medida que avanza y comunicar los resultados a los afectados. Se puede ver un ejemplo en vivo del despliegue en [el GitLab de la iniciativa](#), con el sitio web auto-generado disponible en [sus páginas GitLab](#).



Aquí hay un flujo de trabajo estándar para usar la funcionalidad de despliegue:

1. Bifurque My GGI Board en su propia instancia o proyecto de GitLab y configúrelo siguiendo las instrucciones del fichero README del proyecto: <https://gitlab.ow2.org/ggi/my-ggi-board>. Esto causará lo siguiente:
 - Creará todas las actividades como asuntos/tickets del proyecto.
 - Creará un cuadro para ayudarle a visualizar y gestionar las actividades.
 - Creará un sitio web estático, publicado en las páginas de su instancia de GitLab, con la información extraída de las actividades.
 - Actualizará la descripción del proyecto con los enlaces adecuados al cuadro de actividades y a su sitio web estático.
2. A partir de ahí, puede Ud. empezar a mirar las actividades y rellenar la sección del cuadro de puntuación.
 - La sección del cuadro de puntuación es el equivalente electrónico (y simplificado) de las tarjetas de puntuación ODT mencionados anteriormente. Se emplean para adaptar la actividad a su contexto, listando los recursos locales, riesgos y oportunidades, y definiendo objetivos adaptados necesarios para completar la actividad.
 - Si alguna actividad no se aplica a su contexto, simplemente márkela como "No seleccionada", o ciérrela.
 - Este es un proceso que consume bastante tiempo, pero es muy necesario, ya que le ayudará, paso a paso, a trazar su propia hoja de ruta y plan.
3. Cuando se hayan definido las actividades, puede comenzar a implementar su propia OSPO. Seleccione algunas actividades que considere importantes para empezar, y cambie su etiqueta de avance de 'No iniciada' a 'En Curso'. Puede utilizar las funciones de GitLab para ayudarle a organizar el trabajo (comentarios, encargados, etc.) o cualquier otra herramienta. Es fácil de enlazar con las actividades, y hay un montón de estupendas integraciones disponibles.

4. Periódicamente (semanal, mensualmente, en función de su calendario), evalúe y revise las actividades actuales y, cuando se completen, cambie la etiqueta de "En curso" a "Realizada". Seleccione algunas otras y comience de nuevo en el paso 3 hasta completarlas todas.

El sitio web propone una visión general rápida de las actividades actuales y pasadas, y extrae la sección de cuadro de mando de los tickets para mostrar solo la información relevante a nivel local. Cuando se producen cambios en los asuntos (actividades), estos se actualizan automáticamente en el sitio web generado. Tenga en cuenta que las cadenas de integración continua para la generación automática del sitio web se ejecutan automáticamente todas las noches, pero puede iniciarlas fácilmente desde la sección CI / CD del proyecto GitLab. La siguiente imagen muestra la interfaz del sitio web generada automáticamente.

The screenshot shows a web dashboard for 'My Good Governance Initiative'. At the top, there are navigation links for 'Dashboard' and 'My Board'. The main heading is 'Welcome', followed by a sub-heading 'Current activities' and a link '[details]'. Below this, there is a list of activities with their progress: 17 activities not started, 4 activities in progress, and 4 activities done. The 'Current activities' section lists two specific activities: 'Open source enabling digital transformation (GGI-A-37)' with 50% progress (1 done / 2 total) and 'Support open source communities (GGI-A-30)' with 66% progress (2 done / 3 total). Each activity has a blue progress bar.

Puede preguntar u obtener asistencia para la función de despliegue en nuestra página de inicio de GitLab. Y agradecemos sus comentarios.

Página principal de GGI Deploy: <https://gitlab.ow2.org/ggi/my-ggi-board>

4.5 Disfrute

¡Comunique su éxito y disfrute de la tranquilidad de una estrategia de código abierto de vanguardia!

El Buen Gobierno de OSS es un método para implementar un programa de mejora continua, y como tal nunca termina. Sin embargo, es importante destacar los pasos intermedios y apreciar los cambios que produce, para hacer visible el progreso y compartir los resultados.

- Comuníquese con los afectados y usuarios finales para hacerles conocer las ventajas y beneficios que aporta el esfuerzo de la iniciativa.
- Fomentar la sostenibilidad del programa. Asegúrese de que las mejores prácticas y lecciones aprendidas del programa se apliquen y actualicen siempre.
- Comparta su experiencia con sus colegas: aporte comentarios al grupo de trabajo de GGI y dentro de su comunidad de adopción de OSPO, y comparta su enfoque.

5 Actividades del objetivo de uso

5.1 Inventario de competencias y recursos de código abierto

ID de la Actividad: [GGI-A-17](#).

Descripción

En cualquier etapa, desde el punto de vista de la gestión, es útil tener un mapa, un inventario de los recursos de código abierto, los activos, el uso y su estado, así como las necesidades potenciales y las soluciones disponibles. También incluye la evaluación del esfuerzo y las habilidades necesarias para cubrir la brecha.

Esta actividad tiene como objetivo tomar una instantánea de la situación del código abierto dentro de la organización y en el mercado y evaluar el puente entre ambos.

- Inventario del uso de OSS en la cadena de desarrollo de software, así como en los productos y componentes de software utilizados en la producción.
- Identifique las tecnologías de código abierto (soluciones, marcos de trabajo, características innovadoras) que podrían ajustarse a sus necesidades y ayudar a mejorar su proceso.

No se incluye

- Identificar y calificar los ecosistemas y comunidades de OSS relacionados. (Objetivo Cultura)
- Identificar las dependencias de las bibliotecas y componentes de OSS. (Objetivo Confianza)
- Identificar las habilidades técnicas (por ejemplo, lenguajes, marcos..) y blandas (por ejemplo, colaboración, comunicación) necesarias. (pertenece a las siguientes actividades: Crecimiento de la competencia de OSS y habilidades de desarrollo de software de código abierto)

Evaluación de oportunidades

Un inventario de los recursos de código abierto disponibles que ayudará a optimizar la inversión y a priorizar el desarrollo de competencias.

Esta actividad crea las condiciones para mejorar la productividad del desarrollo, dada la eficacia y la popularidad de los componentes, los principios de desarrollo y las herramientas de OSS, especialmente en el desarrollo de aplicaciones e infraestructuras modernas.

- Esto puede requerir la simplificación de la cartera de recursos de OSS.
- Esto puede requerir una nueva formación del personal.
- Esto permite identificar las necesidades y alimenta su hoja de ruta informática.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Existe una lista viable de recursos de OSS que "utilizamos", "integramos", "producimos", "alojamos" y las habilidades relacionadas
- Estamos en el camino de mejorar la eficiencia utilizando métodos y herramientas de última generación.
- Hemos identificado recursos de OSS no contabilizados hasta ahora (que pueden haberse colado, y ¿tenemos elementos para definir la política en este ámbito?)
- Pedimos a los nuevos proyectos que respalden o reutilicen los recursos de OSS existentes.
- Tenemos una percepción y una comprensión razonablemente seguras del alcance del uso del OSS en nuestra organización.

Herramientas

Hay muchas formas diferentes de establecer dicho inventario. Una forma sería clasificar los recursos de OSS en cuatro categorías:

- OSS que utilizamos: software que utilizamos en producción o en desarrollo

- OSS que integramos: por ejemplo, bibliotecas de OSS que integramos en una aplicación a medida
- OSS que producimos: por ejemplo, una biblioteca que hemos publicado en GitHub o un proyecto de OSS que desarrollamos o al que contribuimos regularmente.
- OSS que alojamos: OSS que ejecutamos para ofrecer un servicio interno como un CRM, GitLab, nexus, etc. Una tabla de ejemplo sería como la siguiente:

Utilizamos	Integramos	Producimos	Alojamos	Habilidades
Firefox, OpenOffice, Postgresql	Biblioteca slf4j	Biblioteca YY en GH	GitLab, Nexus	Java, Python

La misma identificación debe aplicarse a las competencias

- Competencias y experiencias disponibles a través de los equipos existentes
- Habilidades y experiencias que podrían desarrollarse o adquirirse internamente (formación, entrenamiento, experimentación)
- Competencias y experiencias que deben buscarse en el mercado o a través de la asociación/contratación

Recomendaciones

- Mantenga las cosas simples.
- Se trata de un ejercicio de relativo alto nivel, no de un inventario detallado para el departamento de contabilidad.
- Si bien esta actividad es un buen punto de partida, no es necesario completarla al 100% antes de iniciar otras actividades.
- Manejar asuntos, recursos y habilidades relacionadas con el **desarrollo de software** en la actividad #42.
- El inventario debe abarcar todas las categorías de TI: sistemas operativos, middlewares, DBMS, administración de sistemas, herramientas de desarrollo y pruebas, etc.
- Empiece a identificar a las comunidades afines: es más fácil conseguir apoyo y comentarios del proyecto cuando ya le conocen.

Recursos

- Un curso excelente sobre [Software \(/Libre\), y de Código Abierto \(FOSS\)](#), por el profesor Dirk Riehle.

Próximas actividades propuestas

- [GGI-A-18 - Crecimiento de la competencia en código abierto](#) La identificación de habilidades y recursos de código abierto permite a la organización empezar a consolidar y reforzar su concienciación y competencia.
- [GGI-A-19 - Supervisión del código abierto](#) Una vez completado el inventario de software de código abierto y de competencias, se puede empezar a controlar y gestionar el uso de OSS dentro de la organización.
- [GGI-A-28 - Perspectiva de los recursos humanos](#) El departamento de Recursos Humanos puede elaborar planes, contratos y procesos de desarrollo proporcionados y adecuados basándose en el inventario elaborado en esta actividad.
- [GGI-A-33 - Comprometerse con los proveedores de código abierto](#) Antes de definir una relación externa con un proveedor, uno debe conocer su software de código abierto y sus competencias.
- [GGI-A-42 - Gestionar habilidades y recursos de código abierto](#) Una vez completado el inventario de activos y habilidades de código abierto, se puede empezar a gestionarlos adecuadamente, aprovechando los recursos internos existentes.

5.2 Aumento de la competencia de código abierto

ID de la actividad: [GGI-A-18](#).

Descripción

Esta actividad consiste en planificar e iniciar las habilidades técnicas y la experiencia inicial con OSS una vez que se ha realizado un inventario (nº 17). También es la oportunidad de empezar a establecer una hoja de ruta de desarrollo de habilidades básicas y ligeras.

- Determinar cuáles son las habilidades y la formación necesarias.
- Establezca un proyecto piloto para poner en marcha el enfoque, aprenda de la práctica y fije un primer hito.
- Aprovechar la experiencia adquirida y crear un corpus de conocimientos.
- Empezar a identificando y documentando los próximos pasos para una adopción más amplia.
- Elaborar una estrategia en los próximos meses o un año para conseguir apoyo financiero y de gestión.

El alcance de la actividad:

- Linux, Apache, Debian, habilidades de administración.
- Bases de datos de código abierto MariaDB, MySQL, PostgreSQL, etc.
- Tecnologías de virtualización y nube de código abierto.
- Pila LAMP y sus alternativas.

Evaluación de oportunidades

Como cualquier tecnología informática, y probablemente aún más, el código abierto aporta innovación. El código abierto crece rápido y cambia con rapidez. Requiere que las organizaciones se mantengan al día.

Esta actividad ayuda a identificar las áreas en las que la formación podría ayudar a las personas a ser más eficientes y sentirse más seguras utilizando el código abierto. Ayuda a tomar decisiones sobre el desarrollo de los empleados. Sembrar habilidades básicas de código abierto permite evaluar la oportunidad de:

- Ampliar las soluciones informáticas con tecnologías de mercado existentes desarrolladas por el ecosistema.
- Desarrollar nuevas formas de colaboración dentro y fuera de la organización.
- Adquirir competencias en tecnologías nuevas e innovadoras.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Se elabora una matriz de competencias.
- El alcance de las tecnologías de OSS utilizadas se define de forma proactiva, es decir, se evita el uso incontrolado de tecnologías de OSS.
- Se adquiere un nivel satisfactorio de pericia para estas tecnologías.
- Los equipos han recibido una formación de "fundamentos del código abierto" para empezar.

Herramientas

Una herramienta clave en este sentido es la denominada matriz (o mapa) de actividades (o competencias).

Esta actividad puede ser realizada por:

- utilizando tutoriales en línea (muchos gratuitos en Internet),
- participando en congresos de desarrolladores,
- recibir formación de proveedores, etc.

Recomendaciones

- Utilizar y desarrollar componentes de código abierto de forma segura y eficiente requiere una mentalidad abierta y colaborativa que debe reconocerse y propagarse tanto desde arriba (gerencia) como desde abajo (desarrolladores).

- Asegúrese de que la dirección apoya y promueve activamente el enfoque. No pasará nada si no hay compromiso por parte de la jerarquía.
- Implique a la gente (desarrolladores, partes interesadas) en el proceso: organizar mesas redondas y escuchar ideas.
- Conceda tiempo y recursos para que la gente descubra, pruebe y juegue con estos nuevos conceptos. Si es posible, que sea divertido: la gamificación y las recompensas son buenos incentivos.

Un proyecto piloto con los siguientes pasos podría servir de catalizador:

- Identificar la tecnología o el marco de trabajo con el que empezar.
- Encuentre formación en línea, tutoriales y ejemplos de código para experimentar.
- Construir un prototipo de la solución final.
- Identificar a algunos expertos a quienes desafiar y entrenar en la implementación.

Recursos

- [Qué es una matriz de competencias](#): una rápida lectura introductoria.
- [Cómo hacer una matriz de competencias para su equipo](#): una plantilla con comentarios.
- [MOOC sobre cultura libre](#) (sólo en francés): se trata de un curso de 6 partes sobre cultura libre, introducción a los derechos de autor, propiedad intelectual y licencias de código abierto

Próximas actividades propuestas

- [GGI-A-28 - Perspectiva de los recursos humanos](#) Planificar cuidadosamente el conjunto de habilidades internas y el crecimiento de las competencias disponibles forma parte de la perspectiva de los recursos humanos.

5.3 Supervisión del código abierto

ID de la Actividad : [GGI-A-19](#).

Descripción

Esta actividad consiste en controlar el uso del código abierto y garantizar una gestión proactiva del software de código abierto. Esto afecta a varias perspectivas, ya sea utilizar herramientas y soluciones empresariales de OSS, o incluir OSS como componentes en desarrollos propios o modificar una versión de un software adaptándolo a las propias necesidades, etc. También se trata de identificar las áreas en las que el código abierto se ha convertido en una solución de facto (a veces encubierta) y evaluar su idoneidad.

Puede que sea necesario aclarar lo siguiente:

- ¿Se proporciona la funcionalidad requerida?
- ¿Se proporciona funcionalidad adicional innecesaria pero que aumenta la complejidad de las fases de CONSTRUCCIÓN y EJECUCIÓN?
- ¿Qué exige la licencia? ¿Cuáles son las limitaciones legales?
- ¿En qué medida la decisión hace a su organización independiente de los proveedores?
- ¿Existe una opción de asistencia preparada para las necesidades de su empresa y cuánto cuesta?
- TCO (Coste Total de Propiedad).
- ¿Conoce la dirección las ventajas del código abierto, por ejemplo, más allá del "ahorro de costes de licencia"? Sentirse cómodo con el código abierto ayuda a sacar el máximo provecho de la colaboración con las comunidades y los proveedores de los proyectos .
- Ver si tiene sentido compartir los costes de desarrollo cediendo los desarrollos propios a la comunidad y todas sus implicaciones, como el cumplimiento de las licencias.
- Compruebe la disponibilidad de soporte comunitario o profesional.

Evaluación de oportunidades

Definir un proceso de decisión específicamente dirigido al código abierto es una forma de maximizar sus beneficios.

- Evita el uso progresivo incontrolado y los costes ocultos de las tecnologías OSS.
- Conduce a decisiones estratégicas y organizativas informadas y conscientes del OSS.

Costes: la actividad puede cuestionar y reconsiderar el uso subóptimo de facto del código abierto por ineficiente, arriesgado, etc.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- El OSS se ha convertido en una opción cómoda a la hora de seleccionar nuevo software.
- El OSS no se considera una excepción ni una opción peligrosa.
- El OSS se ha convertido en una opción "corriente".
- Los actores clave están suficientemente convencidos de que la solución de código abierto tiene ventajas estratégicas en las que merece la pena invertir.
- Puede demostrarse que el coste total de propiedad de la solución basada en código abierto aporta a su organización un valor superior al de la alternativa.
- Se evalúa cómo la independencia de los proveedores ahorra dinero o puede ahorrarlo en el futuro.
- Se evalúa que la independencia de la solución reduce los riesgos de que sea demasiado costoso cambiar la solución (no es posible utilizar formatos de datos cerrados).

Herramientas

En este momento, no podemos pensar en ninguna herramienta relevante o afectada por esta actividad.

Recomendaciones

- Gestionar de forma proactiva el uso del código abierto requiere unos niveles básicos de concienciación y comprensión de los fundamentos del código abierto, ya que deben tenerse en cuenta en cualquier decisión sobre OSS.
- Compare la funcionalidad necesaria en lugar de buscar una alternativa para una solución de código cerrado conocida.
- Asegúrese de contar con apoyo y un desarrollo activo.
- Tenga en cuenta los efectos de la licencia de la solución en su organización.
- Convencer a todos los agentes clave del valor de las ventajas del código abierto, más allá del "ahorro de costes de licencia".
- Sea honesto, no exagere el efecto de la solución de código abierto.
- En el proceso de toma de decisiones es igualmente importante evaluar las distintas soluciones de código abierto para evitar decepciones por expectativas erróneas, dejar claro lo que se pide a la organización y todas las ventajas que aporta la apertura de las soluciones. Esto debe identificarse para que la organización pueda evaluarlo en función de su propio contexto.

Recursos

- [Las 5 principales ventajas del código abierto](#): Blog patrocinado, pero aún así interesante, lectura rápida.
- [Weighing The Hidden Costs Of Open Source](#): un estudio patrocinado por IBM sobre los costes de soporte del OSS.

5.4 Software empresarial de código abierto

ID de la Actividad: [GGI-A-20](#).

Descripción

Esta actividad trata de seleccionar proactivamente soluciones de OSS, soportadas por proveedores o por la comunidad, en áreas orientadas a negocio. Puede también abarcar la

definición de políticas de preferencia para la selección de aplicaciones de código abierto en negocios.

Aunque el software de código abierto es el más utilizado por los profesionales de TI -sistema operativo, middleware, DBMS, administración de sistemas, herramientas de desarrollo-, aún tiene que ser reconocido en áreas donde los profesionales de la empresa son los principales usuarios.

La actividad abarca áreas como: software ofimático, entornos colaborativos, administración de usuarios, gestión de flujos de trabajo, gestión de relaciones con clientes, correo electrónico, comercio electrónico, etc.

Evaluación de oportunidades

A medida que el uso de código abierto se normaliza, llega mucho más allá de los sistemas operativos y herramientas de desarrollo. Paulatinamente va encontrando su camino hacia las capas superiores de los sistemas de información, incluyendo las aplicaciones de negocio. Es importante identificar qué aplicaciones de OSS se usar con éxito para atender las necesidades de la organización y como convertirse en la elección preferida para ahorrar costes en una organización.

La actividad puede conllevar algunos costes de formación y migración.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Existe una lista de soluciones OSS recomendadas para abordar necesidades pendientes en aplicaciones comerciales.
- Se ha esbozado una política de preferencia para la selección de software de aplicaciones de negocio de código abierto.
- Se están evaluando las aplicaciones de negocio privativas en uso contra sus equivalentes OSS.
- El proceso de compras y las solicitudes de ofertas especifican la preferencia por el código abierto (si es legalmente viable).

Herramientas

En esta etapa, no podemos pensar en ninguna herramienta relevante o implicada en esta actividad.

Recomendaciones

- Hable con colegas, aprenda de lo que hacen otras empresas comparables a la suya.
- Asista a eventos profesionales locales para descubrir más acerca de las soluciones y soporte profesional OSS.
- Experimente primero con las ediciones y soporte comunitarios antes de comprometerse con planes de soporte de pago.

Recursos

- [What is enterprise open source?](#): una lectura rápida sobre código abierto para empresas.
- [101 Open Source Apps to Help your Business Thrive](#): una lista indicativa de soluciones de código abierto para empresas.

Próximas actividades propuestas

- [GGI-A-33 - Relación con proveedores de código abierto](#) Mejore la confianza en sus activos de código abierto relacionándose con profesionales del código abierto.
- [GGI-A-43 - Política de adquisiciones de código abierto](#) El uso de OSS por parte de las empresas se optimizará conociendo los activos que ya existen y teniendo una política de adquisiciones clara al respecto.

5.5 Gestionar competencias y recursos de código abierto

ID de la actividad: [GGI-A-42](#).

Descripción

Esta actividad se centra en las habilidades y en los recursos de **desarrollo de software**. Incluye las tecnologías y habilidades específicas de desarrollo de los desarrolladores, así como de los procesos, métodos y herramientas de desarrollo en general.

Para las tecnologías de código abierto hay disponible una amplia cantidad de documentación, foros y debates que provienen del ecosistema, y recursos públicos. Para beneficiarse plenamente de su enfoque de código abierto, es necesario establecer una mapa de ruta de sus recursos actuales y metas deseadas para establecer un programa consistente de habilidades, métodos y herramientas de desarrollo en los equipos.

Ámbitos de la aplicación

Es necesario establecer los dominios en los que se aplicará el programa, y como irá mejorando la calidad y eficiencia del código y de las prácticas. Como ejemplo, el programa no tendrá los mismos beneficios si solo hay un único programador trabajando en componentes de código abierto que si todo o ciclo de vida del desarrollo se optimiza para incluir las buenas prácticas del código abierto.

Es necesario definir el ámbito de aplicación para el desarrollo de código abierto: componentes técnicos, aplicaciones, modernización o nuevos desarrollos. Algunos ejemplos de prácticas de desarrollo que se podrían beneficiar del código abierto serían:

- Administración de la nube.
- Aplicaciones nativas en la nube; cómo innovar con estas tecnologías.
- DevOps, Integración Continua / Entrega Continua.

Categorías

- Competencias y recursos necesarios para desarrollar software de código abierto: propiedad intelectual, licencias, prácticas.
- Habilidades y recursos necesarios para desarrollar software utilizando componentes, lenguajes y tecnologías de código abierto.
- Habilidades y recursos necesarios para utilizar métodos y procesos de código abierto.

Evaluación de oportunidades

Las herramientas de código abierto son cada vez más populares entre los desarrolladores. Esta actividad aborda la necesidad de evitar la proliferación de herramientas heterogéneas dentro de un equipo de desarrollo. Ayuda a definir una política en este ámbito. Ayuda a optimizar la formación y la adquisición de experiencia. Se emplea un inventario de competencias para la contratación, la formación y la planificación de la sucesión en caso de que un empleado clave abandone la empresa.

Necesitaríamos una metodología para mapear las competencias de desarrollo de software de código abierto.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Hay una descripción de la cadena de producción del software libre (la "cadena de suministro de software"),
- Existe un plan (o una lista de deseos) para la racionalización de los recursos de desarrollo,
- Existe un inventario de competencias en el que se resumen las aptitudes, formación y experiencia de los desarrolladores actuales,
- Existe una lista de deseos en materia de formación y un programa que aborda las carencias de competencias,
- Hay una lista de buenas prácticas de desarrollo de código abierto que faltan y un plan para aplicarlas.

Recomendaciones

- Empiece por lo básico, haga crecer el análisis y la hoja de ruta de forma constante.
- A la hora de contratar, haga especial hincapié en las aptitudes y la experiencia en código abierto. Siempre es más fácil cuando las personas ya tienen un ADN de código abierto que formarlas y entrenarlas.
- Consulte los programas de formación de los proveedores de software y las escuelas de código abierto.

Recursos

Más información:

- Una introducción a [¿qué es un inventario de competencias?](#) de Robert Tanner.
- Un artículo sobre competencias en código abierto: [5 habilidades de código abierto para mejorar su juego y su currículum](#)

Esta actividad puede incluir recursos y competencias técnicas como:

- **Lenguajes populares** (como Java, PHP, Perl, Python).
- **Marcos de trabajo de código abierto** (Spring, AngularJS, Symfony) y herramientas de prueba.
- **Métodos y buenas prácticas de desarrollo** ágil, DevOps y de código abierto.

Próximas actividades propuestas

- [GGI-A-28 - Perspectiva de los Recursos Humanos](#) Una vez identificados internamente los recursos de código abierto para ayudar a la concienciación sobre el código abierto, haga que el departamento de Recursos Humanos también los valore, tanto para los empleados actuales como para los futuros.

6 Actividades del objetivo de confianza

6.1 Gestionar el cumplimiento de la legislación

ID de la Actividad: [GGI-A-21](#).

Descripción

Las organizaciones necesitan implementar un proceso de cumplimiento legal para asegurar su uso y participación en proyectos de código abierto.

Una gestión madura y profesional del cumplimiento legal, en la organización y en toda la cadena de suministro, abarca:

- Realizar un análisis exhaustivo de la propiedad intelectual que incluya la identificación de licencias y su comprobación de compatibilidad.
- Asegurar que la organización puede utilizar, integrar, modificar y redistribuir componentes de código abierto como parte de sus productos o servicios con seguridad.
- Proporcionar a los empleados propios y subcontratados un proceso transparente para crear y contribuir al software de código abierto.

Análisis de composición de software (SCA): una parte importante de los problemas legales y de propiedad intelectual se derivan del uso de componentes publicados bajo licencias que son incompatibles entre ellos o incompatibles con la forma en que la organización desea usar y redistribuir los componentes. SCA es el primer paso para resolver esos problemas, ya que "necesita conocer el problema para solucionarlo". El proceso consiste en identificar todos los componentes involucrados en un proyecto en una Lista de Materiales, incluidas las dependencias de construcción y pruebas.

Comprobación de licencias: Un proceso de comprobación de licencias utiliza una herramienta para analizar automáticamente la base de código e identificar las licencias y los derechos de autor que contiene. Si se ejecuta con regularidad, e idealmente se integra en cadenas de construcción e integración continuas, permite detectar problemas de propiedad intelectual en una fase temprana.

Evaluación de oportunidades

Con el creciente uso de OSS en los sistemas de información de las organizaciones, es esencial evaluar y gestionar la posible exposición legal.

Sin embargo, comprobar las licencias y los derechos de autor puede ser difícil y costoso. Los desarrolladores necesitan ser capaces de comprobar rápidamente las preguntas IP y legales. Contar con un equipo y un responsable corporativo dedicado a cuestiones legales y de propiedad intelectual garantiza una gestión proactiva y coherente de las cuestiones jurídicas, ayuda a asegurar las contribuciones y el uso de los componentes de código abierto y proporciona una visión estratégica clara.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Hay un proceso de verificación de licencias fácil de usar disponible para proyectos.
- Hay un proceso fácil de usar de comprobación de la propiedad intelectual disponible para proyectos.
- Existe un equipo o persona responsable del cumplimiento legal dentro de la organización.
- Se programan auditorías periódicas para evaluar el cumplimiento legal.

Otras formas de establecer puntos de verificación:

- Existe un proceso de verificación de licencias fácil de usar.
- Hay un equipo jurídico/IP fácil de usar.
- Todos los proyectos proporcionan la información necesaria para que las personas utilicen y contribuyan al proyecto.
- Hay un contacto en el equipo para preguntas relativas a la propiedad intelectual y el licenciamiento.

- Hay un funcionario corporativo dedicado a la propiedad intelectual y el licenciamiento.
- Hay un equipo dedicado a cuestiones relativas a la propiedad intelectual y al licenciamiento.

Herramientas

- [ScanCode](#)
- [Fossology](#)
- [SW360](#)
- [Fossa](#)
- [OSS Review Toolkit](#)

Recomendaciones

- Informar al personal sobre los riesgos asociados al licenciamiento conflictivo con los objetivos empresariales.
- Proponga una solución sencilla para que los proyectos establezcan la comprobación de licencias en su base de código.
- Comunique su importancia y ayude a los proyectos a agregarlo a sus sistemas de integración continua.
- Proporcione una plantilla o directrices oficiales para la estructura del proyecto.
- Establezca comprobaciones automáticas para asegurarse de que todos los proyectos cumplen las directrices.
- Considere realizar una auditoría interna para identificar las licencias de la infraestructura de la empresa.
- Proporcione formación básica sobre propiedad intelectual y licencias al menos a una persona por equipo.
- Proporcione formación completa sobre propiedad intelectual y licencias al funcionario corporativo.
- Establezca un proceso para escalar al funcionario corporativo los asuntos relacionados con la propiedad intelectual y las licencias.

Recuerde que el cumplimiento no es sólo una cuestión jurídica, sino también de propiedad intelectual. Así que aquí van algunas cuestiones para ayudar a entender las consecuencias del cumplimiento legal:

- Si distribuyo un componente de código abierto y no respeto las condiciones de la licencia, infrinjo la licencia --> implicaciones legales.
- Si utilizo un componente de código abierto dentro de un proyecto que deseo distribuir/publicar, esa licencia puede obligar a tener visibilidad sobre elementos de código que no deseo abrir --> Impacto de confidencialidad para la ventaja táctica de mi empresa y con terceros (implicaciones legales).
- Es un debate abierto si el uso de una licencia de código abierto para un proyecto que quiero publicar otorga PI relevante --> implicaciones de PI.
- Si hago que un proyecto sea de código abierto *antes* de cualquier proceso de patente, eso *probablemente* excluye la creación de patentes relativas al proyecto --> implicaciones de PI.
- Si hago que un proyecto sea de código abierto *después* de cualquier proceso de patente, eso *probablemente* permita la creación de patentes (defensivas) relativas a ese proyecto --> potencial de PI.
- En proyectos complejos que incorporan muchos componentes con muchas dependencias, la multitud de licencias de código abierto puede presentar incompatibilidades entre licencias --> implicaciones legales (cf. actividad GGI-A-23 - Gestionar las dependencias de software).

Recursos

- Existe una amplia lista de herramientas en la [página existente del grupo de cumplimiento de OSS](#).
- [Prácticas de cumplimiento de código abierto recomendadas para la empresa](#). Un libro de Ibrahim Haddad, de la Fundación Linux, sobre prácticas de cumplimiento de código abierto para la empresa. [Proyecto OpenChain](#)

Próximas actividades propuestas

- [GGI-A-24 - Gestión de indicadores clave](#) Haga visibles y medibles las preocupaciones, los procesos y los resultados relacionados con el cumplimiento de la legislación. Esto ayudará a las personas a darse cuenta de su importancia en una fase más temprana del proceso.

6.2 Gestionar las vulnerabilidades del software

ID de la actividad: [GGI-A-44](#).

Descripción

Nuestro código es tan seguro como su parte menos segura. Casos recientes (por ejemplo, [heartbleed](#)², [equifax](#)³) han demostrado la importancia de verificar vulnerabilidades en partes del código que no desarrolla directamente la entidad. Las consecuencias de las exposiciones van desde fugas de datos (con un tremendo impacto en la reputación) hasta ataques de ransomware e indisponibilidad de servicios que amenaza el negocio.

Se sabe que el software de código abierto tiene una gestión de vulnerabilidades mejor que el software privativo, principalmente porque:

- Hay más ojos buscando problemas y soluciones en código y procesos abiertos.
- Los proyectos de código abierto corrigen vulnerabilidades y publican parches y versiones nuevas mucho más rápidamente.

Por ejemplo, un [estudio de WhiteSource](#) sobre software privativo mostró que el 95% de las vulnerabilidades encontradas en sus componentes de código abierto ya habían publicado una solución en el momento del análisis. La cuestión, por tanto, es **gestionar mejor las vulnerabilidades tanto en el código base como en sus dependencias**, independientemente de que sean de código cerrado o abierto.

Para mitigar estos riesgos, hay que establecer un programa de evaluación de sus activos de software y un proceso de verificación de vulnerabilidad ejecutado con regularidad. Implemente herramientas que alertan a los equipos afectados, gestione vulnerabilidades conocidas y evite amenazas derivadas de las dependencias del software.

Evaluación de oportunidades

Cualquier empresa que use software tiene que vigilar sus vulnerabilidades en:

- su infraestructura (por ejemplo, infraestructura en la nube, infraestructura de red, almacenes de datos),
- sus aplicaciones empresariales (RRHH, herramientas CRM, gestión de datos internos y relacionados con los clientes),
- su código interno: por ejemplo, el sitio web de la empresa, proyectos de desarrollo interno, etc,
- y todas las dependencias directas e indirectas de software y servicios.

El ROI de las vulnerabilidades es poco conocido hasta que ocurre algo malo. Hay que tener en cuenta las consecuencias de una violación de datos importante o de la indisponibilidad de servicios para estimar el verdadero coste de las vulnerabilidades.

Del mismo modo, se debe evitar a toda costa una cultura de secreto y ocultamiento por cuestiones relacionadas con la seguridad dentro de la empresa. En cambio, la información sobre el estado de la vulnerabilidad debe compartirse y debatirse para encontrar las mejores respuestas por parte de las personas adecuadas, desde desarrolladores hasta altos ejecutivos.

Los beneficios de prevenir ciberataques gestionando cuidadosamente las vulnerabilidades del software son múltiples:

- Evite los riesgos reputacionales,

²<https://www.wikipedia.org/wiki/Heartbleed>

³<https://arstechnica.com/information-technology/2017/09/massive-equifax-breach-caused-by-failure-to-patch-two-month-old-bug/>

- Evite mermas en la explotación (DDoS, Ransomware, tiempo para reconstruir un sistema informático alternativo tras un ataque)),
- Cumpla con la normativa de protección de datos.

Gestionar vulnerabilidades de software OSS es sólo una parte del proceso de ciberseguridad más amplio que aborda globalmente la seguridad de los sistemas y servicios en la organización.

Evaluación del progreso

Debe haber una persona o un equipo dedicado a supervisar las vulnerabilidades y procesos fáciles de usar en los que puedan confiar los desarrolladores. La evaluación de vulnerabilidades es una parte estándar del proceso de integración continua, y las personas pueden supervisar el estado actual del riesgo en un panel de control específico.

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- La actividad queda cubierta cuando se evalúan y supervisan todos los programas y servicios internos buscando vulnerabilidades conocidas.
- La actividad queda cubierta cuando se implantan una herramienta y un proceso dedicados en la cadena de producción de software para evitar la introducción de problemas en las rutinas diarias de desarrollo.
- Una persona o equipo es responsable de evaluar el riesgo de la CVE/vulnerabilidad frente a la exposición.
- Una persona o equipo se encarga de enviar la CVE/vulnerabilidad a los afectados (SysOps, DevOps, desarrolladores, etc.).

Herramientas

- Herramientas de GitHub
 - GitHub proporciona pautas y herramientas para proteger el código alojado en la plataforma. Consulte [GitHub docs](#) para obtener más información.
 - GitHub proporciona [Dependabot](#) para identificar vulnerabilidades en dependencias automáticamente.
- [Eclipse Steady](#) es una herramienta gratuita de código abierto que analiza los proyectos Java y Python buscando vulnerabilidades y ayuda a los desarrolladores a mitigarlos.
- [OWASP dependency-check](#): un escáner de vulnerabilidades de código abierto.
- [OSS Review Toolkit](#): un orquestador de código abierto capaz de recopilar avisos de seguridad para dependencias usadas desde servicios de datos de vulnerabilidad configurados.

Recursos

- La [base de datos de CVEs de vulnerabilidades de MITRE](#) . Consulte también la [Base de datos de NVDs de seguridad del NIST](#) , y recursos satélite como [CVE Details](#).
- Consulte también esta nueva iniciativa de Google: las [vulnerabilidades de código abierto](#).
- El grupo de trabajo OWASP publica una lista de escáneres de vulnerabilidades [en su sitio web](#), tanto del mundo comercial como de código abierto.
- J. Williams y A. Dabirsiaghi. La desafortunada realidad de las bibliotecas inseguras, 2012.
- [Detección, evaluación y mitigación de vulnerabilidades en dependencias de código abierto](#), Serena Elisa Ponta, Henrik Plate & Antonino Sabetta, Ingeniería Empírica de Software volumen 25, páginas 3175-3215 (2020).
- [Un conjunto de datos curado manualmente de correcciones a vulnerabilidades de software de código abierto](#), Serena E. Ponta, Henrik Plate, Antonino Sabetta, Michele Bezzi, Cédric Dangremont. También hay un [kit de herramientas en desarrollo para implementar el conjunto de datos mencionado](#).

Próximas actividades propuestas

- [GGI-A-24 - Gestión de indicadores clave](#) Haga visibles las vulnerabilidades identificadas. Esto ayudará a la gente a darse cuenta de lo seguro o no que es su software, y demostrará la importancia de seleccionar las dependencias adecuadas.

6.3 Gestionar las dependencias del software

ID de la Actividad: [GGI-A-23](#).

Descripción

Un programa *identificador de dependencias* busca las dependencias utilizadas realmente en la base de código. Como resultado, la organización debe establecer y mantener una lista de dependencias conocidas de su base de código y observar la evolución de los proveedores identificados.

Establecer y mantener una lista de dependencias conocidas es un elemento facilitador y un requisito previo:

- Control de propiedad intelectual y licencias: algunas licencias no pueden mezclarse, ni siquiera como dependencia. Uno debe conocer sus dependencias para evaluar sus riesgos jurídicos asociados.
- Gestión de vulnerabilidades: todo el software es tan débil como su parte más débil: véase el ejemplo del fallo [Heartbleed](#). Hay que conocer sus dependencias para evaluar los riesgos de seguridad asociados.
- Ciclo de vida y sostenibilidad: una comunidad activa en el proyecto de dependencia es un buen indicio de cara a la corrección de errores, optimizaciones y nuevas funciones.
- Selección meditada de las dependencias utilizadas, según criterios de "madurez": el objetivo es utilizar componentes de código abierto que sean seguros, con una base de código saneada y bien mantenida, y una comunidad viva, activa y reactiva que acepte contribuciones externas, etc.

Evaluación de oportunidades

La identificación y el seguimiento de las dependencias es un paso necesario para mitigar los riesgos asociados a cualquier reutilización de código. Además, implantar herramientas y procesos para gestionar las dependencias de software es un requisito previo para gestionar adecuadamente la calidad, el cumplimiento y la seguridad.

Considere las siguientes preguntas:

- ¿Cuál es el riesgo de la empresa (coste, reputación, etc.) si el software se corrompe o sufre ataques técnicos o legales?
- ¿La base de código se considera crítica para las personas, la organización o el negocio?
- ¿Y si un componente del que depende una aplicación cambia su repositorio?

El primer paso mínimo es implementar una herramienta de análisis de composición de software (SCA). Quizá se requiera el apoyo de consultoras especializadas para un SCA o mapeo de dependencias completo.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Se identifican las dependencias de todo el código desarrollado internamente.
- Se identifican las dependencias de todo el código externo ejecutado en la empresa.
- Existe un procedimiento de análisis de composición de software o de identificación de dependencias fácil de implementar para que los proyectos lo añadan a su proceso de integración continua.
- Se utilizan herramientas de análisis de dependencias.

Herramientas

- [OWASP Dependency check](#): dependency-Check es una herramienta de análisis de composición de software (SCA) que intenta detectar vulnerabilidades de dominio público contenidas en las dependencias de un proyecto.
- [OSS Review Toolkit](#): un conjunto de herramientas para ayudar a revisar las dependencias del software de código abierto.

- [Fossa](#): análisis de dependencias rápido, portable y fiable. Soporta escaneo de licencias y vulnerabilidades. Independiente del lenguaje; se integra con más de 20 sistemas de compilación.
- [Software 360](#).
- [Herramienta de licencia Eclipse Dash](#): toma una lista de dependencias y solicita a [Clearly Defined](#) que compruebe sus licencias.
- [Proyecto FOSSology](#): FOSSology es un proyecto de código abierto cuya misión es avanzar en el cumplimiento de las licencias de código abierto.

Recomendaciones

- Realice auditorías periódicas sobre las dependencias y los requisitos de propiedad intelectual para mitigar los riesgos jurídicos.
- Lo ideal es integrar la gestión de dependencias en el proceso de integración continua para que los problemas (nueva dependencia, incompatibilidad de licencia) se identifiquen y solucionen lo antes posible.
- Realice un seguimiento de las vulnerabilidades relacionadas con las dependencias, mantenga informados a los usuarios y a los desarrolladores.
- Informe a las personas sobre los riesgos asociados con un licenciamiento incorrecto.
- Proponga una solución sencilla para que los proyectos establezcan la comprobación de licencias en su base de código.
- Comunique su importancia y ayude a los proyectos a agregarlo a sus sistemas de integración continua.
- Establezca un KPI visible para los riesgos relacionados con dependencias.

Recursos

- Página existente del grupo [Herramientas para el cumplimiento de licencias OSS](#).
- [Cumplimiento de licencias de software libre y de código abierto: Herramientas para el análisis de la composición del software](#), por Philippe Ombredanne, nexB Inc.
- [Modelo de madurez de la sostenibilidad del software](#).
- [CHAOS](#): Community Health Analytics Open Source Software.

Próximas actividades propuestas

- [GGI-A-21 - Gestionar el cumplimiento legal](#) Antes de poder hacer un seguimiento de las incompatibilidades de la propiedad intelectual y las licencias, es necesario identificar todas las dependencias de su software de código abierto.
- [GGI-A-22 - Gestionar las vulnerabilidades del software](#) Antes de ser capaz de rastrear las vulnerabilidades en sus activos de código, uno necesita identificar todas las dependencias en su software de código abierto.

6.4 Gestionar los indicadores clave

ID de la actividad: [GGI-A-24](#).

Descripción

Esta actividad recopila y supervisa un conjunto de indicadores que informan las decisiones de gestión cotidiana y las opciones estratégicas relativas al software de código abierto gestionado profesionalmente.

Las métricas clave relacionadas con el software de código abierto conforman el contexto para valorar la aplicación de los programas de gobernanza. La actividad abarca la selección de algunos indicadores, la publicación para los equipos y la gestión, y el envío periódico de actualizaciones sobre la iniciativa, por ejemplo mediante un boletín informativo o noticias corporativas.

Esta actividad requiere:

- partes interesadas para debatir y definir los objetivos del programa,

- la implantación de una herramienta de medición y recopilación de datos conectada a la infraestructura de desarrollo,
- la publicación de al menos un cuadro de mando para las partes interesadas y para todas las personas involucradas en la iniciativa.

Los indicadores se basan en datos que deben recopilarse de fuentes pertinentes. Afortunadamente, hay muchas fuentes para la ingeniería de software de código abierto. Por ejemplo:

- el entorno de desarrollo, la cadena CI/CD de producción,
- el departamento de RRHH,
- las herramientas de pruebas y de análisis de composición de software,
- los repositorios.

Algunos ejemplos de indicadores son:

- Número de dependencias resueltas, mostradas por tipo de licencia.
- Número de dependencias obsoletas/vulnerables.
- Número de incidencias con licencias o propiedad intelectual detectadas.
- Contribuciones a proyectos externos.
- Vida de los defectos.
- Número de colaboradores en un componente, número de commits, etc.

Esta actividad consiste en definir estos requisitos y necesidades de medición, e implantar un cuadro de mando que muestre de forma sencilla y eficaz los principales indicadores del programa.

Evaluación de oportunidades

Los indicadores clave ayudan a comprender y gestionar mejor los recursos dedicados al software de código abierto, y miden los resultados con el fin de comunicar de manera efectiva y obtener los beneficios completos de la inversión. Al divulgarse más personas pueden seguir la iniciativa y se sentirán implicadas, lo que lo convierte en una preocupación y un objetivo a nivel de organización.

Si bien cada actividad tiene criterios de evaluación que ayudan a responder a las preguntas sobre los progresos logrados, todavía es necesario supervisar los números e indicadores cuantitativos.

Ya sea en una pequeña startup o una gran empresa global, las métricas clave ayudan a mantener los equipos enfocados y supervisar el rendimiento. Las métricas son cruciales porque apoyan la toma de decisiones y son la base para supervisar las decisiones ya adoptadas.

Con números y gráficos simples y prácticos, todos los miembros de la organización podrán seguir y sincronizar los esfuerzos en materia de código abierto, lo que lo hará una preocupación y acción compartidas. Esto también facilita a diversos actores su incorporación, contribuir al proyecto y obtener los beneficios generales.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Se ha establecido una lista de métricas y cómo recogerlas.
- Se utilizan herramientas para recoger, almacenar, procesar y mostrar indicadores.
- Existe un cuadro de mando general a disposición de todos los participantes que muestra el progreso realizados en la iniciativa.

Herramientas

- [GrimoireLab](#) de Bitergia.
- Las herramientas genéricas de BI (elasticsearch, grafana, visualizaciones R/Python...) también son buenas opciones, cuando los conectores adecuados se configuran de acuerdo con los objetivos definidos.

Recomendaciones

- Anote los objetivos y la hoja de ruta del gobierno del código abierto.

- Comunicar internamente las acciones y el estado de la iniciativa.
- Involucrar al personal en la definición de los indicadores clave, para asegurar que
 - se entienden bien,
 - proporcionan una visión completa de las necesidades y
 - se tengan en cuenta y se sigan.
- Construya al menos un cuadro de mando que pueda mostrarse a todo el mundo (por ejemplo, en una pantalla en la sala), con indicadores esenciales que muestren el progreso y la situación general.

Recursos

- La [comunidad CHAOSS](#) tiene muchas buenas referencias y recursos relacionados con los indicadores de código abierto.
- Consulte las métricas para [Atributos del proyecto](#) de la [metodología](#) de Niveles de Preparación del Mercado de OW2.
- [Una nueva forma de medir la apertura: el índice de gobernanza abierta](#) de Liz Laffan es una lectura interesante sobre la apertura en proyectos de código abierto.
- [Indicadores de Gobernanza: Una Guía de Usuario](#) es la guía de la ONU sobre indicadores de gobernanza. Aunque se aplica a la democracia, la corrupción y la transparencia de las naciones, vale la pena leer los conceptos básicos de medición e indicadores aplicados a la gobernanza.

Próximas actividades propuestas

- [GGI-A-37 - El código abierto al servicio de la transformación digital](#) Utilice las métricas producidas como parte de su estrategia general de código abierto.

6.5 Revisión del código

ID de la Actividad: [GGI-A-44](#).

Descripción

La revisión del código es una tarea rutinaria que implica la revisión manual y/o automatizada del código fuente de una aplicación antes de lanzar un producto o entregar un proyecto al cliente. En el caso del software de código abierto, la revisión del código es algo más que detectar errores de forma oportunista: es un enfoque integrado del desarrollo colaborativo llevado a cabo a nivel de equipo.

Las revisiones del código deben aplicarse tanto al código desarrollado internamente como al reutilizado de fuentes externas, ya que mejora la confianza general en el código y refuerza la propiedad. También es una forma excelente de mejorar las habilidades y conocimientos globales dentro del equipo y fomentar la colaboración en equipo.

Evaluación de oportunidades

Las revisiones del código son valiosas siempre que la organización desarrolle software o reutilice piezas de software externas. Aunque se trata de un paso estándar en el proceso de ingeniería de software, las revisiones de código en el contexto del código abierto aportan ventajas específicas como:

- Al publicar código fuente interno, compruebe que se respetan las directrices de calidad adecuadas.
- Cuando contribuya a un proyecto de código abierto existente, compruebe que se respetan las directrices del proyecto en cuestión.
- La documentación disponible al público se actualiza en consecuencia.

También es una excelente oportunidad para compartir y hacer cumplir algunas de las normas de la política de cumplimiento legal de su empresa, como:

- No elimine nunca las cabeceras de licencia o derechos de autor existentes en el código fuente abierto reutilizado.

- No copies ni pegues código fuente de Stack Overflow sin permiso previo del equipo legal.
- Incluya la línea de copyright correcta cuando sea necesario.

Las revisiones del código aportarán confianza al código. Si las personas no están seguras de la calidad o de los riesgos potenciales de utilizar un producto de software, deben realizar revisiones del código y entre iguales.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- La revisión del código fuente abierto se reconoce como un paso necesario.
- Se planifican revisiones del código fuente abierto (de forma periódica o en momentos críticos).
- Se ha definido y aceptado colectivamente un proceso para realizar revisiones del código fuente abierto.
- Las revisiones de código abierto son una parte habitual del proceso de desarrollo.

Recomendaciones

- La revisión del código es una tarea colectiva que funciona mejor en un buen entorno de colaboración.
- No dude en utilizar las herramientas y patrones existentes en el mundo del código abierto, donde las revisiones de código son un estándar desde hace años (décadas).

Recursos

- [What is Code Review?](#): una lectura didáctica sobre la revisión de código que se encuentra en la Open Practice Library de Red Hat.
- [Best Practices for Code Reviews](#): otra perspectiva interesante sobre lo que es la revisión de código.

Próximas actividades propuestas

- [GGI-A-26 - Contribuir a proyectos de código abierto](#) La revisión del código es una práctica común en los proyectos de código abierto, ya que mejora la calidad del código y el intercambio de conocimientos. Los colaboradores que realizan revisiones del código suelen sentirse más cómodos con las contribuciones externas y la colaboración.

7 Actividades del objetivo de cultura

7.1 Promover las mejores prácticas de desarrollo del código abierto

ID de la Actividad: [GGI-A-25](#).

Descripción

Esta actividad consiste en definir, promover y aplicar activamente las mejores prácticas de código abierto dentro de los equipos de desarrollo.

Como punto de partida se podría considerar atender los siguientes temas:

- Documentación de usuario y desarrollador.
- Organización adecuada del proyecto en un repositorio de acceso público.
- Promover y aplicar la reutilización controlada.
- Proporcionar una documentación completa y actualizada del producto.
- Gestión de la configuración: flujos de trabajo git, patrones colaborativos.
- Gestión de lanzamientos: liberación temprana y frecuente, versiones estables frente a versiones en desarrollo, etc.

Los proyectos de la OSS tienen un modus operandi especial que recuerda a un [bazar](#). Para permitir y fomentar esta colaboración y mentalidad, se recomiendan algunas prácticas que faciliten el desarrollo colaborativo y descentralizado y las contribuciones de desarrolladores externos...

Documentos de la comunidad Asegúrese de que todos los proyectos dentro de la empresa presenten los siguientes documentos:

- README -- descripción rápida del proyecto, cómo interactuar, enlaces a recursos.
- Contributing -- introducción para personas que desean contribuir.
- Código de conducta -- comportamiento considerado aceptable, o no, en la comunidad.
- LICENSE -- la licencia por omisión del repositorio.

REUSE -- buenas prácticas para reutilizar [REUSE](#) es una iniciativa de la [Free Software Foundation Europe](#) para mejorar la reutilización del software y simplificar el cumplimiento de las licencias.

Evaluación de oportunidades

Aunque depende en gran medida del conocimiento común del OSS entre el equipo, capacitar a las personas y crear procesos que hagan cumplir estas prácticas es siempre beneficioso. Es aún más importante cuando:

- se desconocen los posibles usuarios y contribuyentes,
- los desarrolladores no están acostumbrados al desarrollo de código abierto.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- El proyecto establece una lista de buenas prácticas de código abierto que hay que cumplir.
- El proyecto supervisa su adecuación a las buenas prácticas.
- El equipo de desarrollo se ha concienciado para cumplir con las buenas prácticas de OSS.
- Se evalúan periódicamente buenas prácticas nuevas y se procura aplicarlas.

Herramientas

- La herramienta [REUSE helper tool](#) ayuda a un repositorio a cumplir las buenas prácticas de [REUSE](#). Puede incluirse en muchos procesos de desarrollo para confirmar el estado actual.
- [ScanCode](#) tiene la capacidad de listar todos los documentos comunitarios y legales del repositorio: véase su [descripción de características](#).

- GitHub tiene una característica estupenda para [ver los documentos comunitarios ausentes](#). Se puede encontrar en la página del repositorio > "Insights" > "Community".

Recomendaciones

- La lista de buenas prácticas depende del contexto y dominio del programa y se debe revisar periódicamente al estilo de la mejora continua. Las prácticas deben vigilarse y evaluarse periódicamente para seguir el progreso.
- Formar al personal en reutilización (como consumidores) del OSS y en ecosistemas (como contribuyentes).
- Implementar REUSE.software como en la actividad #14.
- Establecer un proceso para gestionar los riesgos jurídicos asociados con la reutilización y las contribuciones.
- Alentar al personal a contribuir a proyectos externos.
- Proporcione una plantilla o directrices oficiales para la estructura del proyecto.
- Establezca comprobaciones automáticas para asegurarse de que todos los proyectos cumplen las directrices.

Recursos

- [La lista de buenas prácticas de código abierto de OW2](#) de la metodología de evaluación de los Niveles de Preparación del mercado.
- [El sitio web oficial de REUSE](#) con especificación, tutorial y preguntas frecuentes.
- [Directrices de la comunidad de GitHub](#).
- Un ejemplo de [buenas prácticas de gestión de la configuración utilizando GitHub](#).

Próximas actividades propuestas

- [GGI-A-42 - Gestión de habilidades y recursos de código abierto](#) Puede añadir la lista de mejores prácticas de desarrollo de código abierto identificadas al material de formación general.
- [GGI-A-44 - Revisión del código](#) La revisión del código es un elemento esencial de las mejores prácticas de desarrollo.

7.2 Contribuir a los proyectos de código abierto

ID de la actividad: [GGI-A-26](#).

Descripción

Contribuir a proyectos de código abierto que se utilizan libremente es uno de los principios fundamentales del buen gobierno. El punto es evitar ser un simple consumidor pasivo y devolver a los proyectos. Cuando la gente agrega una característica o arregla un defecto para su propio fin, deben hacerlo lo suficientemente genérico para contribuir al proyecto. Los desarrolladores deben tener tiempo para las contribuciones.

Esta actividad abarca el siguiente ámbito:

- Trabajar en origen con proyectos de código abierto.
- Notificar defectos y solicitar funciones.
- Contribuir con código y corrección de defectos.
- Participar en las listas de correo de la comunidad.
- Compartir experiencias.

Evaluación de oportunidades

Los principales beneficios de esta actividad son:

- Aumenta el conocimiento general y el compromiso con el código abierto dentro de la empresa, ya que la gente empieza a contribuir y a implicarse en proyectos de código abierto. Consiguen una sensación de utilidad pública y mejoran su reputación personal.

- La empresa aumenta su visibilidad y reputación a medida que las contribuciones se abren camino a través del proyecto contribuido. Esto demuestra que la empresa está realmente implicada en el código abierto, contribuye y promueve la equidad y la transparencia.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Existe una vía clara y oficial para las personas que desean contribuir.
- Se anima a los desarrolladores a contribuir a los proyectos de código abierto que utilizan.
- Existe un proceso para garantizar el cumplimiento y la seguridad jurídicos de las contribuciones de los desarrolladores.
- KPI: Volumen de contribuciones externas (código, listas de correo, incidencias..) por individuo, equipo o entidad.

Herramientas

Puede ser útil hacer seguimiento de las contribuciones, tanto para llevar la cuenta de lo que se aporta como para poder comunicar el esfuerzo de la empresa. Para ello pueden utilizarse cuadros de mando y programas informáticos para el seguimiento de actividad. Compruébe:

- [GrimoireLab](#) de Bitergia
- [ScanCode](#)

Recomendaciones

Animar al personal de la entidad a contribuir a proyectos externos:

- Permitiéndoles invertir tiempo para escribir correcciones de errores y funciones genéricas y bien probadas, y contribuir las a la comunidad.
- Capacitando al personal para contribuir a las comunidades de código abierto. Se trata tanto de las habilidades técnicas (mejorando el conocimiento de su equipo) como de la comunidad (pertenencia a las comunidades de código abierto, código de conducta, etc.).
- Proporcione formación sobre cuestiones jurídicas, de propiedad intelectual y técnicas, y establezca un contacto dentro de la empresa para ayudar con estos temas si la gente tiene dudas.
- Incentivar la publicación de trabajos.
- Tenga en cuenta que las contribuciones de la empresa/entidad reflejarán la calidad de su código y su implicación, así que asegúrese de que su equipo de desarrollo proporciona un código lo suficientemente bueno.

Recursos

- La iniciativa [CHAOSS](#) de la Fundación Linux tiene algunas herramientas e indicaciones sobre cómo hacer seguimiento de las contribuciones al desarrollo.

Próximas actividades propuestas

- [GGI-A-31 - Reafirmar públicamente el uso del código abierto](#) Ahora que existe una contribución y un compromiso públicamente visibles por parte de la organización, ¡empieza a comunicarlo!
- [GGI-A-24 - Gestión de indicadores clave](#) Hacer visible y mensurable la contribución a los proyectos de OSS. Esto ayudará a difundir la iniciativa y a elevar la moral de la gente.
- [GGI-A-27 - Pertenecer a la comunidad de software libre](#) Contribuir a la comunidad de software libre es el primer paso para formar parte de ella. Una vez que la gente empieza a contribuir, se involucra más en la salud y la gobernanza del proyecto y puede llegar a convertirse en mantenedor, garantizando un proyecto y una hoja de ruta sostenibles y saludables.
- [GGI-A-29 - Implicarse en proyectos de código abierto](#) Los proyectos de código abierto valoran la meritocracia. Ahora que has demostrado un buen conocimiento del código y los procesos, puedes implicarte en el proyecto y oficializar tus contribuciones.

- [GGI-A-36 - El código abierto como factor de innovación](#) Contribuir a proyectos de OSS e interactuar con colaboradores externos es un factor de fomento de la innovación.
- [GGI-A-39 - Upstream primero](#) Contribuir a proyectos de OSS realmente tiene sentido si las actualizaciones están disponibles en el proyecto upstream, de forma regular e institucionalizada.

7.3 Pertener a la comunidad de código abierto

ID de la Actividad: [GGI-A-27](#).

Descripción

Esta actividad trata de desarrollar entre los desarrolladores un sentimiento de pertenencia a la gran comunidad del código abierto. Como en toda comunidad, las personas y entidades tienen que participar y contribuir al conjunto. Refuerza los vínculos entre los profesionales y aporta sostenibilidad y actividad al ecosistema. Desde un punto de vista más técnico, permite elegir las prioridades y la hoja de ruta de los proyectos, mejora el nivel de conocimientos generales y la conciencia técnica.

Esta actividad abarca lo siguiente:

- **Identifique eventos** a los que merezca la pena asistir. Conectar a la gente, aprender sobre nuevas tecnologías y crear una red son factores clave para obtener todos los beneficios del código abierto.
- Considere la posibilidad de **afiliarse a fundaciones**. Las fundaciones y organizaciones son un componente clave del ecosistema del código abierto. Proporcionan recursos técnicos y organizativos a los proyectos, y son un buen lugar neutral para que los patrocinadores debatan problemas y soluciones comunes, o elaboren estándares.
- Siga a **grupos de trabajo**. Los grupos de trabajo son espacios neutrales de trabajo colaborativo donde los expertos interactúan en torno a un dominio específico como IoT, modelado o ciencia. Son un mecanismo muy eficiente para abordar conjuntamente las preocupaciones comunes y específicas de cada dominio.
- **Participación presupuestaria**. Al final del camino, el dinero es el facilitador. Planifique los gastos necesarios, conceda a las personas tiempo remunerado para estas actividades, prevea los próximos movimientos, para que el programa no tenga que detenerse tras unos meses con estrés de caja.

Evaluación de oportunidades

El código abierto funciona mejor cuando se hace en relación con la comunidad de código abierto en general. Facilita la corrección de errores, el intercambio de soluciones, etc.

También es una buena forma de que las empresas muestren su apoyo a los valores del código abierto. Comunicar la implicación de la empresa es importante tanto para su reputación como para el ecosistema del código abierto.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Se esboza una lista de actos a los que se podría asistir.
- Hay un seguimiento de las charlas públicas que dan los miembros del equipo.
- Las personas pueden presentar solicitudes de participación en eventos.
- La gente puede presentar proyectos para su patrocinio.

Recomendaciones

- Encueste a la gente para saber qué actos les gustan o serían más beneficiosos para su trabajo.
- Establezca comunicaciones internas (boletín, centro de recursos, invitaciones...) para que la gente conozca las iniciativas y pueda participar.

- Asegúrese de que estas iniciativas puedan beneficiar a varios tipos de personas (desarrolladores, administradores, soporte...), no sólo a los altos ejecutivos.

Recursos

- [¿Qué motiva a un desarrollador a contribuir al software de código abierto?](#) Un artículo de Michael Sweeney en clearcode.cc.
- [Por qué las empresas contribuyen al código abierto](#) Un artículo de Velichka Atanasova de VMWare.
- [Por qué sus empleados deberían contribuir al código abierto](#) Una buena lectura de Robert Kowalski de CloudBees.
- [7 maneras en que su empresa puede apoyar el código abierto](#) Un artículo de Simon Phipps para InfoWorld.
- [Eventos: la fuerza vital del código abierto](#) Un artículo de Donna Benjamin de RedHat.

Próximas actividades propuestas

- [GGI-A-28 - Perspectiva de RRHH](#) Si la organización pertenece a la comunidad OSS, es más fácil atraer a personas cualificadas, en función de la comunidad en la que se participe.
- [GGI-A-31 - Afirmar públicamente el uso del código abierto](#) Ahora que ya formas parte de la comunidad del OSS, ¡dalo a conocer! Es bueno para tu reputación, y es bueno para el proyecto en términos de salud y difusión.

7.4 Perspectiva de los recursos humanos

ID de la actividad: [GGI-A-28](#).

Descripción

El cambio a la cultura del código abierto tiene profundas repercusiones en los recursos humanos:

- **Nuevos procesos y contratos:** Los contratos tienen que adaptarse para permitir y promover las contribuciones externas. Esto incluye cuestiones de propiedad intelectual y licencias para el trabajo realizado dentro de la empresa, pero también la posibilidad de que el empleado o contratista tenga sus propios proyectos.
- **Diferentes tipos de personas:** Las personas que trabajan con código abierto suelen tener incentivos y mentalidades diferentes a los de las personas puramente propietarias y corporativas. Los procesos y las mentalidades tienen que adaptarse a este paradigma orientado a la reputación de la comunidad, para atraer a nuevos tipos de talento y retenerlos a lo largo del tiempo.
- **Desarrollo profesional:** es necesario ofrecer una trayectoria profesional que fomente y valore las habilidades técnicas y sociales de los empleados, así como las competencias que espera su organización (colaboración para impulsar los esfuerzos de la comunidad, comunicación para actuar como portavoz de su empresa, etc.). Por supuesto, los Recursos Humanos tienen un papel clave a la hora de hacer posible el código abierto como objetivo cultural.

Fuerza laboral Para un desarrollador que ha estado trabajando en la misma solución privativa durante mucho tiempo, cambiar a código abierto puede parecer un gran cambio y requiere una adaptación. Pero para la mayoría de los desarrolladores, el software de código abierto solo acarrea beneficios.

Todos los desarrolladores que salen hoy de la escuela o de la universidad han estado siempre trabajando en código abierto. Dentro de una empresa, la gran mayoría de los desarrolladores utilizan lenguajes de código abierto e importan bibliotecas o fragmentos de código abierto todos los días. De hecho, es mucho más fácil pegar líneas de código fuente abierto en un programa que desencadenar el proceso de provisión interna, que escala a través de múltiples aprobaciones a lo largo de la línea de mando.

El código abierto hace que el trabajo del desarrollador sea más interesante, porque con el código abierto, un desarrollador siempre está pendiente de lo que han inventado sus compañeros de fuera de la empresa y en consecuencia se mantiene a la vanguardia de la tecnología.

Para una organización, tiene que haber una estrategia de Recursos Humanos para 1/ cualificar o recualificar a la mano de obra existente 2/ reflejar y posicionar a la empresa en la contratación de nuevos talentos, de ahí cuál es el atractivo de la empresa en lo que respecta al código abierto.

“Obtener gente con una buena mentalidad FLOSS, que ya entienden el código, y saber cómo trabajar bien con otros es maravilloso. La alternativa de evangelizar / formar / entrenar vale la pena pero es más cara en dinero y tiempo.”

— Director ejecutivo de un proveedor de software OSS

Esto ilustra que la contratación de personas con ADN de código abierto es una vía de aceleración a tener en cuenta en la estrategia de los Recursos Humanos.

Procesos

- Establecer o revisar las descripciones de los puestos de trabajo (aptitudes técnicas, aptitudes interpersonales, competencias y experiencias)
- Programas de capacitación: autodidacta, formación formal, entrenamiento acompañado en gestión, mapeo de colegas, comunidades
- Establecer o revisar la trayectoria profesional: competencias, resultados/impacto clave y etapas de la carrera

Evaluación de oportunidades

1. Enmarcar las prácticas de desarrollo: probablemente el problema no sea tanto incitar a los desarrolladores a utilizar más código abierto, sino asegurarse de que lo utilizan de forma segura, respetando las condiciones de licencia de cada tecnología de código abierto y sin renunciar a los controles de seguridad tradicionales (las líneas de código abierto podrían contener código malicioso),
2. Revisar las prácticas de colaboración: con las prácticas de desarrollo, la oportunidad es extender la agilidad y la colaboración a otras líneas de negocio de su organización. A menudo se recurre al InnerSource para fomentar estos comportamientos, aunque esto sea solo la mitad del camino hacia la cultura del código abierto,
3. Cultura de la organización: al final, todo depende de la cultura de su organización: el código abierto puede ser el buque insignia de valores como la apertura, la colaboración, la ética o la sostenibilidad.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Hay formación disponible para presentar tanto las ventajas como las restricciones (cumplimiento de los términos de licencia de propiedad intelectual) relativas al código abierto.
- Cada desarrollador, cada arquitecto, cada jefe de proyecto (o responsable de producto/negocio), entiende los beneficios y las restricciones (cumplimiento de los términos de licencia de propiedad intelectual) relativos al código abierto.
- Se anima a los desarrolladores a contribuir a las comunidades de código abierto y a responsabilizarse de ellas, y podrían recibir la formación adecuada para hacerlo.
- Las cualificaciones y competencias se reflejan en las descripciones de los puestos de trabajo de la organización y en el escalafón profesional.
- La experiencia que los desarrolladores adquieren en código abierto (contribuciones a comunidades de código abierto, participación en el proceso de cumplimiento interno, portavoces externos de la empresa, ...) se tiene en cuenta en el proceso de evaluación de los Recursos Humanos.

Herramientas

- Matriz de competencias.
- Programas de formación pública (ej. escuela de código abierto).
- Abastecimiento: GitHub, GitLab, LinkedIn, Meetups, Epitech, Epita...
- Modelos de contrato (Cláusula de fidelidad).
- Descripciones de puestos (plantillas) y etapas profesionales (plantillas).

Recomendaciones

Hoy en día, la mayoría de los desarrolladores ya conocen algunos principios del código abierto y están dispuestos a trabajar con y en software de código abierto. Sin embargo, todavía hay algunas medidas que la dirección debería tomar:

- Preferencia por la experiencia en OSS a la hora de contratar, aunque el trabajo para el que se contrata al desarrollador sólo esté relacionado con tecnología privativa. Lo más probable es que, con la transformación digital, el desarrollador tenga que trabajar algún día sobre código abierto.
- Programa de formación en OSS: Cada desarrollador, cada arquitecto, cada jefe de proyecto (o responsable de producto/negocio), debe tener acceso a recursos de formación (vídeos o formación presencial) que presenten las ventajas del código abierto y también las restricciones en términos de cumplimiento de licencias y propiedad intelectual.
- Debe ofrecerse formación a los desarrolladores que quieran contribuir a las comunidades de código abierto y formar parte de los órganos de gobierno de estas comunidades (certificaciones Linux).
- Reconocimiento en los procesos de evaluación personal de los Recursos Humanos de la contribución del empleado (desarrollador o arquitecto) a temas relacionados con el código abierto, como las contribuciones a comunidades de código abierto y el cumplimiento de los términos de licencia de propiedad intelectual. La mayoría de los temas son compartidos y encajan en las trayectorias profesionales técnicas, mientras que algunos podrían o deberían ser específicos.
- El secreto mejor guardado y la postura de la empresa: necesidad de abordar los aspectos de la comunicación (cuán fundamental es esto para su organización que podría reflejarse en su informe anual), cómo afecta su postura de comunicación (un colaborador de código abierto podría ser un portavoz de su empresa, incluidos los contactos de prensa).

Recursos

- En cuanto a la capacidad de las personas para hablar en actos fuera de la empresa, véase la Actividad 31: "(Objetivo de compromiso) Afirmar públicamente el uso del código abierto".

7.5 Cambio en origen

ID de la Actividad: [GGI-A-39](#).

Descripción

Esta actividad se ocupa de crear conciencia con respecto a los beneficios de contribuir y hacer cumplir el primer principio de cambio en origen.

Con el enfoque de cambios en origen, todo desarrollo en un proyecto de código abierto se hará con el nivel de calidad y apertura requerido para presentarlo a su núcleo de desarrolladores y que lo publiquen.

Evaluación de oportunidades

Escribir código pensando en el origen produce:

- código de mejor calidad,
- código que está listo para presentarlo en origen,
- código que se integra con el software original,
- código que será compatible con la versión futura,
- reconocimiento por parte de la comunidad del proyecto y una cooperación mejor y más rentable.

Cambios en Origen es más que "ser amable". Significa que Ud. tiene una opinión en el proyecto. Significa predictibilidad. Significa que Ud. tiene el control. Significa que Ud. actúa en lugar de reaccionar. Significa que Ud. entiende la fuente abierta.

— [Maximilian Michels](#)

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad: ¿Se implementan los cambios en origen?

- Incremento significativo del número de solicitudes de integración presentadas a proyectos de terceros.
- Se ha esbozado una lista de proyectos de terceros para los que se deben aplicar los cambios en origen.

Recomendaciones

- Identificar a los desarrolladores más expertos interactuando con sus homólogos originales.
- Facilitar que los desarrolladores interactúen con sus homólogos originales (eventos, hackatones, etc.)

Recursos

- Una explicación clara del principio de Cambios en Origen y por qué se corresponde con el objetivo de Cultura: <https://maximilianmichels.com/2021/upstream-first/>.

Cambios en Origen significa que cada vez que Ud. resuelve un problema en su copia del código original de la que otros podrían beneficiarse, contribuye estos cambios al origen. Es decir, solicita su integración con el repositorio original.

- [¿Qué significa "aguas arriba" o "aguas abajo" en un contexto de desarrollo de software?](#) Una explicación cristalina.
- Explicado en los documentos de diseño del sistema operativo Chromium: [Upstream First](#).
- Red Hat acerca del origen y los beneficios de [cambiar en origen](<https://www.redhat.com/en/blog/what-open-source-upstream>).

Próximas actividades propuestas

- [GGI-A-25 - Promover las buenas prácticas en el desarrollo del código abierto](#) Contribuir a las fuentes es una de las mejores prácticas del código abierto. Inclúyala también en las buenas prácticas de la organización, ya que contribuirá a las aportaciones externas, a la calidad general interna y al intercambio de conocimientos.

8 Actividades del objetivo de compromiso

8.1 Participar en proyectos de código abierto

ID de la actividad: [GGI-A-29](#).

Descripción

Esta actividad consiste en comprometerse a realizar contribuciones significativas a algunos proyectos de OSS que sean importantes para usted. Las contribuciones se escalan y se comprometen a nivel de organización (no a nivel personal como en nº 26). Pueden adoptar diversas formas, desde la financiación directa hasta la asignación de recursos (por ejemplo, personas, servidores, infraestructura, comunicación, etc.), siempre que beneficien al proyecto o ecosistema de forma sostenible y eficiente.

Esta actividad es una continuación de la actividad nº 26 y lleva las contribuciones de los proyectos de código abierto al nivel de la organización, haciéndolas más visibles, potentes y beneficiosas. En esta actividad, se supone que las contribuciones aportan una mejora sustancial y a largo plazo al proyecto de OSS: por ejemplo, un desarrollador o equipo que desarrolla una nueva característica muy deseada, activos de infraestructura, servidores para un nuevo servicio, asunción del mantenimiento de una rama muy utilizada.

La idea es destinar un porcentaje de los recursos a patrocinar a desarrolladores de código abierto que escriben y mantienen bibliotecas o proyectos que utilizamos.

Esta actividad implica tener un mapa del software de código abierto utilizado, y una evaluación de su criticidad para decidir cuál apoyar.

Evaluación de oportunidades

Si cada empresa que utiliza código abierto contribuyera al menos un poco, tendríamos un ecosistema sano. <https://news.ycombinator.com/item?id=25432248>

Apoyar proyectos ayuda a garantizar su sostenibilidad y facilita el acceso a la información, e incluso puede ayudar a influir y priorizar algunos desarrollos (aunque ésta no debería ser la razón principal para apoyar proyectos).

Posibles beneficios de esta actividad: asegurar que se da prioridad a los informes de defecto y que los desarrollos se integran en la versión estable. Posibles costes asociados a la actividad: dedicar tiempo y/o dinero a los proyectos.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Proyecto beneficiario identificado.
- Opción de apoyo decidida, como por ejemplo contribución monetaria directa o contribución de código.
- Se ha nombrado al responsable de la tarea.
- Ha habido alguna contribución.
- Se ha evaluado el resultado de la contribución.

Puntos de verificación tomados del cuestionario de OpenChain [autocertificación](#):

- Tenemos una política de contribución a proyectos de código abierto en nombre de la organización.
- Disponemos de un procedimiento documentado que regula las contribuciones al código abierto.
- Disponemos de un procedimiento documentado para que todo el personal de software conozca la política de contribución al código abierto.

Herramientas

Algunas organizaciones ofrecen mecanismos para financiar proyectos de código abierto (podría ser conveniente si su proyecto objetivo está en sus carteras).

- [Colectivo Abierto](#).
- [Software Freedom Conservancy](#).
- [Tidelift](#).

Recomendaciones

- Concéntrese en los proyectos críticos para la organización: son los proyectos a los que más desea ayudar con sus contribuciones.
- Elija proyectos comunitarios.
- Esta actividad requiere una familiaridad mínima con un proyecto objetivo.

Recursos

- [Cómo apoyar ahora proyectos de código abierto](#): Una breve página con ideas para financiar proyectos de código abierto.
- [Sustain OSS: un espacio de debate sobre el mantenimiento del código abierto](#)

Próximas actividades propuestas

- [GGI-A-26 - Contribuir a proyectos de código abierto](#) La forma más natural de comprometerse con una iniciativa de código abierto es contribuir directamente al proyecto. A cambio, recibirás valiosos comentarios sobre tus aportaciones.
- [GGI-A-30 - Apoyo a las comunidades de código abierto](#) Hay muchas formas de apoyar las iniciativas de código abierto que son esenciales para su organización. Comprometerse con las comunidades es una buena forma de descubrirlas y fomentarlas.

8.2 Apoyar a las comunidades de código abierto

ID de la actividad: [GGI-A-30](#).

Descripción

Esta actividad consiste en relacionarse con representantes institucionales del mundo del código abierto.

Se consigue mediante:

- Afiliación a fundaciones de OSS (incluido el coste económico de la afiliación).
- Apoyando y abogando por las actividades de las fundaciones.

Esta actividad implica asignar a los equipos de desarrollo y TI algo de tiempo y presupuesto para participar en comunidades de código abierto.

Evaluación de oportunidades

Las comunidades de código abierto están a la vanguardia de la evolución del ecosistema del código abierto. Comprometerse con las comunidades de código abierto tiene varias ventajas:

- ayuda a mantenerse informado y al día,
- mejora el perfil de la organización,
- la afiliación conlleva ventajas,
- proporciona estructura y motivación adicionales al equipo informático de código abierto.

Los costes incluyen:

- cuotas de afiliación,
- tiempo del personal y algún presupuesto para viajes asignado para participar en actividades comunitarias,
- seguimiento del compromiso de propiedad intelectual.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- La organización es miembro signatario de una fundación de código abierto.
- La organización participa en la gobernanza.
- El software desarrollado por la organización se envía / se ha añadido a la base de código de una fundación.
- La afiliación se reconoce en los sitios web tanto de la organización como de la comunidad.
- Evaluación de costes y beneficios de la afiliación.
- Se ha designado un punto de contacto para la comunidad.

Recomendaciones

- Únase a una comunidad compatible con su tamaño y sus recursos, es decir, una comunidad que pueda oír su voz y en la que pueda ser un colaborador reconocido.

Recursos

- Consulte esta [página útil](#) de la Fundación Linux sobre por qué y cómo unirse a una comunidad de código abierto.

Próximas actividades propuestas

- [GGI-A-31 - Afirmar públicamente el uso del código abierto](#) Ahora que apoyas oficialmente a algunas comunidades de OSS, ¡dalo a conocer! Es bueno para tu reputación, y es bueno para los proyectos en términos de salud y difusión.

8.3 Hacer público el uso del código abierto

ID de la actividad: [GGI-A-31](#).

Descripción

Esta actividad consiste en reconocer el uso de OSS en un sistema de información, en aplicaciones y en nuevos productos.

- Proporcionar historias de éxito.
- Presentaciones en eventos.
- Financiación de la participación en actos.

Evaluación de oportunidades

En la actualidad se acepta generalmente que la mayoría de los sistemas de información funcionan con OSS, y que las nuevas aplicaciones se hacen en su mayor parte reutilizando OSS.

El principal beneficio de esta actividad es crear igualdad de condiciones entre el OSS y el software privativo, para asegurarse de que se presta la misma atención al OSS y se gestiona con la misma profesionalidad que el software privativo.

Un beneficio secundario es que contribuye en gran medida a elevar el perfil del ecosistema de OSS y, dado que los usuarios de OSS son identificados como "innovadores", también aumenta el atractivo de la organización.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Los proveedores comerciales de código abierto tienen autorización para utilizar el nombre de la organización como referencia del cliente.
- Los colaboradores pueden hacerlo y expresarse bajo el nombre de la organización.
- El uso de OSS se menciona abiertamente en el informe anual del departamento de TI.

- No hay ningún obstáculo para que la organización explique su uso del OSS en los medios de comunicación (entrevistas, eventos sobre OSS y el sector, etc.).

Recomendaciones

- El objetivo de esta actividad no es que la organización se convierta en un organismo de activismo de OSS, sino asegurarse de que no haya ningún obstáculo para que el público reconozca su uso de OSS.

Recursos

- Ejemplo del [CERN](#) afirmando públicamente su uso de OpenStack

8.4 Comprometerse con los proveedores de código abierto

ID de la actividad: [GGI-A-33](#).

Descripción

Asegúrese contratos con proveedores de software de código abierto que proporcionen software crítico para usted. Las empresas y entidades que producen software de código abierto necesitan prosperar para proporcionar mantenimiento y desarrollo de nuevas funciones. Su experiencia específica es necesaria en el proyecto, y la comunidad de usuarios confía en su actividad y contribuciones continuas.

La colaboración con proveedores de código abierto adopta varias formas:

- Suscripción de planes de ayuda.
- Contratación de empresas locales de servicios.
- Desarrollos patrocinadores.
- Pagar por una licencia comercial.

Esta actividad implica considerar los proyectos de código abierto como productos completos por los que merece la pena pagar, al igual que por cualquier producto privativo, aunque normalmente mucho menos caro.

Evaluación de oportunidades

El objetivo de esta actividad es garantizar un apoyo profesional al software de código abierto utilizado en la organización. Tiene varias ventajas:

- Continuidad del servicio mediante la oportuna corrección de defectos.
- Rendimiento del servicio gracias a una instalación optimizada.
- Aclaración de la situación legal/comercial del software utilizado.
- Acceso a información temprana.
- Previsión presupuestaria estable.

El coste es obviamente el de los planes de soporte seleccionados. Otro coste podría ser dejar de subcontratar en masa a grandes integradores de sistemas en favor de una contratación detallada con PYME expertas.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- El código abierto utilizado en la organización cuenta con soporte comercial.
- Se han contratado planes de soporte para algunos proyectos de código abierto.
- El coste de los planes de soporte de código abierto es una partida legítima en el presupuesto de TI.

Recomendaciones

- Siempre que sea posible, busque PYME locales expertas.
- Cuidado con los grandes integradores de sistemas que revenden conocimientos de terceros (revenden planes de asistencia que en realidad ofrecen PYMES expertas en código abierto).

Recursos

Un par de enlaces que ilustran la realidad comercial del software de fuente abierta:

- [Una lectura rápida para entender el código abierto comercial.](#)

8.5 Política de adquisición de código abierto

Identificativo de actividad: [GGI-A-43](#).

Descripción

Esta actividad consiste en implementar un proceso para seleccionar, adquirir, y comprar logicales o servicios de código abierto. También se trata de considerar el coste real del logicial de código abierto, y prepararse para ello. El logicial de código abierto puede ser «gratuito» a primera vista, pero no está exento de costes internos y externos, como integración, formación, mantenimiento, y soporte.

Dicha política exige, que, tanto las soluciones de código abierto, como las propietarias, sean tomadas simétricamente a la hora de evaluar la relación calidad-precio, en tanto que combinación óptima del coste total de propiedad, y calidad. Por ende, el departamento de compras de TI debe considerar de forma activa, y equitativa, las alternativas de código abierto, garantiendo, a la vez, que las soluciones propietarias sean puestas en pie de igualdad en las decisiones de compra.

La preferencia por el código abierto puede afirmarse explícitamente basándose en la flexibilidad intrínseca de la opción de código abierto, cuando no existe una diferencia significativa de coste global, entre las soluciones propietarias, y las de código abierto.

Los departamentos de adquisiciones tienen, que entender, que las empresas, que brindan soporte a los logicales de código abierto suelen carecer de músculo comercial para participar en concursos de adquisiciones y adaptar sus políticas y procesos de adquisición de código abierto en consecuencia.

Evaluación de oportunidades

Varias razones justifican los esfuerzos por establecer políticas específicas de adquisición de código abierto:

- El suministro de software y servicios comerciales de código abierto está creciendo y no puede ignorarse, y requiere la aplicación de políticas y procesos de adquisición específicos.
- Hay una oferta creciente de soluciones comerciales de código abierto altamente competitivas para sistemas de información empresariales.
- Incluso después de adoptar un componente logicial de código abierto gratuito, e integrarlo en una aplicación, hay que proporcionar recursos internos, o externos, para mantener ese código fuente.
- Con las soluciones FOSS, el Coste Total de Propiedad (TCO) suele ser inferior (aunque no necesariamente): no pagar licencias al comprar/actualizar, mercado abierto para proveedores de servicios, opción de proveer, uno mismo, una parte, o la totalidad de la solución.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Las convocatorias de ofertas nuevas solicitan de manera proactiva candidaturas de código abierto.

- El departamento de compras tiene una forma de evaluar las soluciones de código abierto frente a las propietarias.
- Se ha implantado, y documentado, un proceso simplificado de adquisición de logiciales, y servicios, de código abierto.
- Se ha definido, y documentado, un proceso de aprobación basado en la experiencia interfuncional.

Recomendaciones

- «Aseguraos de aprovechar la experiencia de vuestros equipos de TI, DevOps, ciberseguridad, gestión de riesgos, y adquisiciones, al crear el proceso.» (de [5 Open Source Procurement Best Practices](#), en inglés).
- La legislación sobre competencia puede exigir que no se mencione, específicamente, «código abierto».
- Seleccionad la tecnología primero, y, a continuación, solicitad propuestas de servicios de personalización, y asistencia.

Recursos

- [Decision factors for open source software procurement](#): no es nuevo, pero sigue siendo una gran lectura de nuestros colegas de OSS-watch en el Reino Unido. Echad un vistazo a las [diapositivas](#).
- [5 Open Source Procurement Best Practices](#): un artículo reciente sobre la contratación de código abierto, con consejos útiles.

Próximas actividades propuestas

- [GGI-A-33 - Comprometerse con los proveedores de código abierto](#) La definición de una política de documentación le ayuda a identificar a los proveedores y comunidades de OSS que deben importarle y con los que debe comprometerse.

9 Actividades del objetivo de estrategia

9.1 Establecer una estrategia para la gobernanza corporativa del código abierto

ID de la Actividad: [GGI-A-16](#).

Descripción

Definir una estrategia de alto nivel para la gobernanza del código abierto dentro de la empresa garantiza la coherencia y visibilidad de los enfoques tanto hacia el uso interno como hacia las contribuciones y participación externas. Hace que la comunicación de la empresa sea más eficaz al ofrecer una visión y un liderazgo claros y establecidos.

El cambio hacia el código abierto conlleva numerosas ventajas, así como algunas obligaciones y un cambio en la cultura de la empresa. Puede repercutir en los modelos de negocio e influir en la forma en que una organización presenta su valor y su oferta, y en su posición frente a sus clientes y competidores.

Esta actividad incluye las siguientes tareas:

- Establecer un responsable de OSS, con el patrocinio y respaldo de la dirección (superior).
- Establezca y publique una hoja de ruta clara para el código abierto, con objetivos claros y beneficios esperados.
- Asegúrese de que todos los altos directivos la conocen y actúan de acuerdo con ella.
- Promover el OSS dentro de la empresa: animar a la gente a utilizarlo, fomentar las iniciativas internas y el nivel de conocimientos.
- Promover el OSS fuera de la empresa: mediante declaraciones y comunicaciones oficiales, y una participación visible en iniciativas de OSS.

Definir, publicar y hacer cumplir una estrategia clara y coherente también ayuda a que todo el personal de la empresa la acepte y facilita otras iniciativas de los equipos.

Evaluación de oportunidades

Es un buen momento para trabajar en esta actividad si:

- No existe un esfuerzo coordinado por parte de la dirección, y el código abierto sigue considerándose una solución ad hoc.
- Ya existen iniciativas internas, pero no penetran hasta los niveles superiores de dirección.
- La iniciativa se puso en marcha hace algún tiempo, pero se enfrenta a numerosos obstáculos y sigue sin dar los resultados esperados.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- La empresa cuenta con una declaración clara de intenciones respecto al gobierno del código abierto. La declaración debe contener:
 - lo que hay que conseguir,
 - para quién lo hacemos,
 - cuál es el poder del estratega o estrategas y qué no.
- Una hoja de ruta del código abierto está ampliamente disponible y aceptada en toda la empresa.

Recomendaciones

- Establezca un grupo de personas y procesos para definir y supervisar la gobernanza del código abierto en la empresa. Asegúrese de que existe un compromiso claro de la alta dirección con las iniciativas de código abierto.
- Comunicar la estrategia de código abierto dentro de la organización, convertirla en una preocupación importante y en un verdadero compromiso corporativo.
- Garantizar que la hoja de ruta y la estrategia sean bien comprendidas por todos, desde los equipos de desarrollo hasta el personal de gestión e infraestructura.

- Comunicar sus avances, para que la gente sepa en qué punto se encuentra la organización con respecto a su compromiso. Publique periódicamente actualizaciones e indicadores.

Recursos

- [Lista de control y referencias para la gobernanza abierta.](#)
- [L'open source comme enjeu de souveraineté numérique, por Cédric Thomas, CEO de OW2, Taller en Orange Labs, París, 28 de enero de 2020 \(sólo en francés\).](#)
- [Una serie de guías para gestionar el código abierto en la empresa, por la Fundación Linux.](#)
- [Un buen ejemplo de documento estratégico de código abierto, por el grupo LF Energy](#)

Próximas actividades propuestas

- [GGI-A-35 - Código abierto y soberanía digital](#) Una estrategia adecuada para la gobernanza corporativa del código abierto debería mejorar el código abierto y la soberanía digital. Ahora es un buen momento para definir las en el contexto de la organización.
- [GGI-A-34 - C-Level awareness](#) Será necesario el compromiso de los ejecutivos de nivel C para implementar adecuadamente la estrategia corporativa de código abierto. Educarles e implicarles es un buen paso para lograr el éxito.

9.2 Conciencia de la gerencia

ID de la actividad: [GGI-A-34](#).

Descripción

La iniciativa de código abierto de la organización sólo dará sus beneficios estratégicos si se aplica en sus niveles más altos integrando el ADN de código abierto en la estrategia y el trabajo interno de la empresa. Ese compromiso no puede suceder sin si los altos ejecutivos o la gerencia. La formación y la mentalidad de código abierto también deben extenderse a quienes configuran las políticas, decisiones y estrategia general, tanto hacia adentro como hacia afuera de la empresa.

Este compromiso garantiza que las mejoras prácticas, los cambios de mentalidad y las nuevas iniciativas se cumplan con un apoyo constante, benevolente y sostenible de la jerarquía, generando una participación más ferviente de los trabajadores. Modela cómo los actores externos ven la organización, aportando beneficios reputacionales y de ecosistema. También es un medio para establecer la iniciativa y sus beneficios a medio y largo plazo.

Evaluación de oportunidades

Esta actividad se vuelve esencial si/cuando:

- La organización ha establecido objetivos globales relevantes para la gestión del código abierto, pero se debate por alcanzarlos. Es improbable que la iniciativa pueda lograr nada sin buenos conocimientos y un claro compromiso por parte de los altos ejecutivos.
- La iniciativa ya ha comenzado y progresa, pero los niveles superiores de la jerarquía no hacen un seguimiento adecuado.

Con suerte, deberá hacerse evidente que todo menos el uso ad hoc de la fuente abierta requiere un enfoque coherente y bien pensado, dado el rango de equipos y cambio cultural que puede traer.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Existe una oficina/funcionario de gobierno con mandato, para establecer una estrategia uniforme de código abierto, en toda la empresa, y garantizar, que el ámbito sea claro.
- Hay un compromiso claro y vinculante de la jerarquía con la estrategia para el código abierto.
- Hay una comunicación transparente por parte de la jerarquía acerca de su compromiso con el programa.

- La jerarquía está disponible para hablar acerca del software de código abierto. Se la puede requerir y confrontar con sus promesas.
- Hay un presupuesto y financiación adecuados para la iniciativa.

Recomendaciones

Los ejemplos de acciones asociadas a esta actividad incluyen:

- Formar a la gerencia para desmitificar el código abierto.
- Obtener un respaldo explícito y práctico para la estrategia y uso del código abierto.
- Mencionar y respaldar explícitamente el programa de código abierto en comunicaciones internas.
- Mencionar y respaldar explícitamente el programa de código abierto en comunicaciones públicas.

El código abierto es un *facilitador estratégico* que embarca a la *cultura empresarial*. ¿Qué significa esto?

- El código abierto puede aprovecharse como mecanismo para desbaratar a los proveedores y reducir los costes de adquisición de logicales.
 - ¿Debería el código abierto ser competencia de los *gestores de activos de software* o de los *departamentos de compras*?
- Las licencias de código abierto consagran las libertades que aportan las ventajas del código abierto, pero también conllevan *obligaciones*. Si no se cumplen adecuadamente, las responsabilidades pueden crear riesgos legales, comerciales y de imagen para una organización.
 - ¿Concederán las condiciones de la licencia visibilidad a recursos que deberían permanecer confidenciales?
 - ¿Afectará eso la cartera de patentes de mi organización?
 - ¿Cómo se debe formar y apoyar a los equipos de proyecto en esta materia?
- Contribuir de vuelta a proyectos externos de código abierto es donde reside el mayor valor del código abierto.
 - ¿Cómo debería mi empresa fomentar (y seguir) esto?
 - ¿Cómo deben utilizar los programadores GitHub, GitLab, Slack, Discord, Telegram, o cualquiera de las demás herramientas que utilizan, habitualmente, los proyectos de código abierto?
 - ¿Puede el código abierto afectar a las políticas de recursos humanos de la empresa?
- Por supuesto, no se trata solo de contribuir, ¿qué hay de mis propios proyectos de código abierto?
 - ¿Estoy listo para hacer innovación *abierta*?
 - ¿Cómo gestionarán mis proyectos las contribuciones *entrantes*?
 - ¿Debo invertir esfuerzo en nutrir una comunidad para un proyecto dado?
 - ¿Cómo debo dirigir la comunidad, qué papel deben tener los miembros de la comunidad?
 - ¿Estoy listo para ceder las decisiones de la hoja de ruta a una comunidad?
 - ¿Puede la fuente abierta ser una herramienta valiosa para reducir el aislamiento entre los equipos de la empresa?
 - ¿Necesito manejar la transferencia de código abierto de una entidad a otra?

Próximas actividades propuestas

- [GGI-A-31 - Afirmar públicamente el uso del código abierto](#) Los ejecutivos de nivel C son representantes destacados de una organización. Haz que comuniquen la implicación de la organización en el código abierto.

9.3 Fuente abierta y soberanía digital

ID de la Actividad: [GGI-A-35](#).

Descripción

La soberanía digital puede definirse como

“La capacidad y la oportunidad de individuos e instituciones para ejercer su papel en el mundo digital de forma independiente, intencionada y segura.” & mdash; Centro de Competencia para las TIC Públicas, Alemania

Para gobernar adecuadamente su negocio, cualquier entidad tiene que depender de otros socios, servicios, productos y herramientas. La revisión de los vínculos y limitaciones de estas dependencias permite a la organización evaluar y controlar su dependencia de factores externos, mejorando así su autonomía y resiliencia.

Como ejemplo, el cautiverio respecto a un proveedor es un factor fuerte de dependencia que puede obstaculizar los procesos de la organización y el valor añadido y, como tal, debe evitarse. La fuente abierta es una de las maneras de salir de este bloqueo. La fuente abierta desempeña un papel importante en la soberanía digital, permitiendo una mayor elección entre soluciones, proveedores e integradores, y un mayor control sobre las hojas de ruta de TI.

Cabe señalar que la soberanía digital no es cuestión de confianza: obviamente necesitamos confiar en nuestros socios y proveedores, pero la relación mejora si se basa en el consentimiento mutuo y el reconocimiento, en lugar de en contratos forzados y restricciones.

He aquí algunas ventajas de una mejor soberanía digital:

- Mejorar la capacidad de la organización para tomar sus propias decisiones sin limitaciones.
- Mejorar la resiliencia de la empresa con respecto a los actores y factores externos.
- Mejorar la posición de negociación al tratar con socios y proveedores de servicios.

Evaluación de oportunidades

- ¿Cuánto cuesta (dificultad y coste) abandonar una solución?
- ¿Podrían los proveedores de soluciones imponer condiciones no deseadas a su servicio (por ejemplo, cambio de licencia, actualizaciones de contratos)?
- ¿Podrían los proveedores de soluciones incrementar unilateralmente sus precios, simplemente porque no tenemos elección?

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta Actividad:

- Hay una evaluación de dependencias críticas respecto a proveedores y socios de la organización.
- Hay una contingencia para estas dependencias identificadas.
- Existe un requisito expreso de soberanía digital cuando se investigan nuevas soluciones.

Recomendaciones

- Identificar los principales riesgos de dependencia de proveedores de servicios y terceros.
- Mantener una lista de alternativas de código abierto a los servicios críticos.
- Añadir un requisito al seleccionar nuevas herramientas y servicios a utilizar dentro de la entidad, indicando la necesidad de soberanía digital.

Recursos

- [A Primer on Digital Sovereignty & Open Source: part I](#) y [A Primer on Digital Sovereignty & Open Source: part II](#), del sitio web Open-Sourcerers .
- Un excelente artículo de [superuser.openstack.org](#) sobre [El papel de la fuente abierta en la soberanía digital](#). He aquí un extracto corto:
La Soberanía Digital es una preocupación clave para el siglo XXI, especialmente para Europa. La fuente abierta tiene un papel importante que desempeñar para habilitar la soberanía digital, permitiendo a todos acceder a la tecnología necesaria, pero también proporcionando la transparencia de gobierno e interoperabilidad necesarias para que las soluciones tengan éxito.
- La postura de la Unión Europea sobre soberanía digital, del [Observador de Fuentes Abiertas \(OSOR\)](#): Fuente abierta, soberanía digital e interoperabilidad: La Declaración de Berlín.
- La postura de UNICEF's sobre la [Soberanía Digital de Fuentes Abiertas](#).

9.4 El código abierto hace posible la innovación

Identificativo de actividad: [GGI-A-36](#).

Descripción

“La innovación es la aplicación práctica de las ideas que dan lugar a la introducción de nuevos bienes o servicios o a la mejora de la oferta de bienes o servicios.”

— Schumpeter, Joseph A.

El código abierto puede ser un factor clave para la innovación a través de la diversidad, la colaboración, y el intercambio fluido de ideas. Personas de distintos entornos, y ámbitos, pueden tener perspectivas diferentes, y aportar respuestas nuevas, mejoradas o incluso disruptivas a problemas conocidos. Se puede hacer posible la innovación escuchando distintos puntos de vista, y promoviendo activamente la colaboración abierta en proyectos y temas.

Del mismo modo, participar en la elaboración, e implementación de normas abiertas es un gran promotor de buenas prácticas, e ideas para mejorar el trabajo diario de la empresa. También permite a la entidad impulsar, e influir en la innovación hacia donde, y en lo, que necesita, así como mejora la visibilidad, y la reputación mundiales.

A través de la innovación, el código abierto permite, no sólo transformar los bienes, o servicios, que vuestra empresa comercializa, sino también crear, o modificar, todo el ecosistema en el que la empresa quiere prosperar.

Por ejemplo, al publicar Android como código abierto, Google invita a cientos de miles de empresas a crear sus propios servicios basados en esta tecnología de código abierto. Google crea, así, todo un ecosistema del que pueden beneficiarse todos los participantes. Por supuesto, muy pocas empresas son lo suficientemente poderosas, como para crear un ecosistema por decisión propia. Pero hay muchos ejemplos de alianzas, entre empresas, para crear un ecosistema de este tipo.

Evaluación de oportunidades

Es importante evaluar la posición de vuestra empresa comparada a la de los competidores, socios, y clientes, porque a menudo sería arriesgado para una empresa alejarse demasiado de las normas y tecnologías utilizadas por clientes, socios, y competidores. Evidentemente, innovar significa ser diferente, pero lo que difiere no debe constituir un ámbito demasiado amplio; de lo contrario, la empresa no se beneficiará de los desarrollos de logicales realizados por las demás empresas del ecosistema, ni del impulso empresarial, que este proporciona.

Evaluación del progreso

Los siguientes **puntos de verificación** demuestran el progreso en esta actividad:

- Se han identificado las tecnologías — y las comunidades de código abierto, que las desarrollan —, que afectan la empresa.
- Los avances, y publicaciones, de estas comunidades de código abierto son objeto de seguimiento — incluso yo conozco la estrategia de ellas, antes de que se hagan públicos los lanzamientos.
- Los empleados de la organización son miembros de (algunas de) estas comunidades de código abierto, e influyen en sus hojas de ruta, y decisiones técnicas, aportando líneas de código, y participando en los órganos de gobierno de dichas comunidades.

Recomendaciones

De todas las tecnologías necesarias, para dirigir vuestra empresa, debéis identificar:

- las tecnologías, que podrían ser las mismas que las de vuestros competidores,
- las tecnologías, que deben ser específicas de vuestra empresa.

Manteneos al día sobre las tecnologías emergentes. El código abierto ha estado impulsando la innovación durante la última década, y muchas herramientas potentes de todos los días proceden de ahí (pensad en Docker, Kubernetes, los proyectos de Big Data de Apache o Linux). No hay

necesidad de saber todo sobre todo, pero uno debe saber lo suficiente del estado de la técnica, para identificar nuevas tendencias interesantes.

Permitid, y animad, a la gente a presentar ideas innovadoras, y a sacarlas adelante. Si es posible, destinad recursos a estas iniciativas, y hacedlas crecer. Confiad en la pasión, y la voluntad de la gente para crear, y fomentar ideas, y tendencias emergentes.

Recursos

- [4 innovations we owe to open source.](#)
- [The Innovations of Open Source](#), del profesor Dirk Riehle.
- [Open source technology, enabling innovation.](#)
- [Can Open Source Innovation Work in the Enterprise?.](#)
- [Europe: Open source software strategy.](#)
- [Europe: Open source software strategy 2020-2023.](#)

9.5 El código abierto al servicio de la transformación digital

Identificativo de la actividad: [GGI-A-37](#).

Descripción

«La Transformación Digital es la adopción de tecnología digital, para transformar servicios, o empresas, mediante la sustitución de procesos no digitales, o manuales, por procesos digitales, o la sustitución de tecnología digital antigua, por tecnología digital reciente.» (Wikipedia)

Cuando las organizaciones más avanzadas en Transformación Digital impulsan conjuntamente el cambio, a través de su negocio, TI, y finanzas, para anclar lo digital por el camino, reconsideran su:

- Modelo de negocio: cadena de valor con ecosistemas, como servicio, SaaS.
- Finanzas: opex/capex, personal, externalización.
- TI: innovación, modernización de legados/activos.

El código abierto está en el centro de la transformación digital:

- Tecnologías, prácticas ágiles, gestión de productos.
- Personas: colaboración, comunicación abierta, ciclo de desarrollo/decisión.
- Modelos de negocio: probar antes de comprar, innovación abierta.

En términos de competitividad, los procesos más visibles son, probablemente, los, que repercuten directamente en la experiencia del cliente. Tenemos, que reconocer, que los grandes actores, así como las empresas emergentes, al ofrecer una experiencia al cliente totalmente sin precedentes, han cambiado, drásticamente, las expectativas de los ellos.

La experiencia del cliente, así como todos los demás procesos de una empresa, dependen, totalmente, de las TI. Todas las empresas tienen que transformar sus TI, y de eso trata la transformación digital. Las empresas que aún no lo han hecho, tienen ahora que lograr su transformación digital, lo más rápido posible, de contrario corren el riesgo de ser eliminadas del mercado. La transformación digital es una condición para la supervivencia. Dado que es mucho lo que está en juego, una empresa no puede dejar por completo la transformación digital en manos de un proveedor. Toda empresa tiene que ponerse manos a la obra con sus TI, lo que significa, que toda empresa tiene que ponerse manos a la obra con el software de código abierto porque no hay TI sin software de código abierto.

En los beneficios esperados de la transformación digital se incluye:

- Simplificar, automatizar los procesos centrales, convertirlos en procesos en tiempo real.
- Permitir respuestas rápidas, a los cambios de la competencia.
- Sacad partido de la inteligencia artificial, y los macrodatos.

Evaluación de oportunidades

La transformación digital podría gestionarse por:

- Segmentos de la TI: TI de producción, TI de apoyo al negocio (gestión de relaciones con el cliente, facturación, compras...), TI de apoyo (recursos humanos, finanzas, contabilidad...), Macrodatos.
- Tipo de tecnología o proceso que soporta la TI: Infraestructura (nube), Inteligencia Artificial, Procesos (Make-or-Buy, DevSecOps, SaaS).

Inyectar código abierto en un segmento o tecnología concretos de vuestra TI revela que queréis entrar de lleno en este segmento o tecnología, porque habéis evaluado, que este segmento, o tecnología concretos de vuestra TI son importantes para la competitividad de vuestra empresa. Es importante evaluar la posición de la empresa en comparación, no sólo, con las de los competidores, sino también con otras industrias, y actores clave en términos de experiencia del cliente, y soluciones de mercado.

Evaluación del progreso

- Nivel 1: Evaluación de la situación

He identificado:

- los segmentos de TI, que son importantes para la competitividad de mi empresa, y
- las tecnologías de código abierto necesarias para desarrollar aplicaciones en estos segmentos.

Y así he decidido:

- sobre qué segmentos quiero gestionar internamente el desarrollo de proyectos, y
- en qué tecnologías de código abierto necesito adquirir experiencia interna.

- Nivel 2: Participación

En algunas tecnologías de código abierto utilizadas en la empresa, varios desarrolladores han recibido formación, y son reconocidos como valiosos colaboradores, por la comunidad de código abierto. En algunos segmentos determinados, se han puesto en marcha proyectos basados en tecnologías de código abierto.

- Nivel 3: Generalización

En todos los proyectos, se estudia, sistemáticamente, una alternativa de código abierto, durante la fase inicial del proyecto. Para facilitar al equipo del proyecto el estudio de dicha alternativa de código abierto, se dedica un presupuesto central, y un equipo central de arquitectos, alojados en el Departamento de TI, a prestar asistencia a los proyectos.

KPIs:

- KPI 1. Cociente para el que se investigó una alternativa de código abierto: (Número de proyectos / Número total de proyectos).
- KPI 2. Cociente por el que se eligió la alternativa de código abierto: (Número de proyectos / Número total de proyectos).

Recomendaciones

Más allá del titular, la Transformación Digital es una mentalidad, que implica algunos cambios fundamentales, y esto también (o incluso principalmente) deberá provenir de las capas superiores de la organización. La dirección debe promover iniciativas, nuevas ideas, gestionar los riesgos, y, potencialmente, actualizar los procedimientos existentes, para adaptarlos a los nuevos conceptos.

La pasión es un gran factor de éxito. Uno de los medios desarrollados por los principales actores en este campo es la creación de espacios abiertos a nuevas ideas, donde la gente pueda presentar, y trabajar libremente sus ideas sobre la transformación digital. La dirección debe fomentar este tipo de iniciativas.

Recursos

- [Fundación Eclipse: Habilitar la transformación digital en Europa mediante la colaboración global del código abierto.](#)
- [Europe: Open source software strategy.](#)
- [Europe: Open source software strategy 2020-2023.](#)

10 Conclusión

Como hemos dicho antes, el buen gobierno del código abierto no es un destino; es un viaje. Tenemos que preocuparnos por nuestros bienes comunes, por las comunidades y el ecosistema que lo hacen posible, porque de ello depende nuestro propio éxito común y, por tanto, individual.

Nosotros, como profesionales del software y entusiastas del código abierto, nos comprometemos a seguir mejorando el manual de la Iniciativa para el Buen Gobierno y a trabajar en su difusión y alcance. Creemos firmemente que las organizaciones, los individuos y las comunidades deben trabajar codo con codo para construir un mejor y mayor conjunto de bienes comunes, disponibles y beneficiosos para todos.

Usted es bienvenido a unirse a la Alianza OSPO, contribuir a nuestro trabajo, correr la voz y ser el embajador de una mejor concienciación y gobernanza del código abierto dentro de su propio ecosistema. Hay una gran variedad de recursos disponibles, desde entradas de blog y artículos de investigación hasta conferencias y cursos de formación en línea. También ofrecemos un conjunto de material útil en [nuestro sitio web](#), y estamos dispuestos a ayudar en todo lo que podamos.

¡Definamos y construyamos juntos el futuro de la Iniciativa para el Buen Gobierno!

10.1 Contacte con

La forma preferida de ponerse en contacto con la Alianza OSPO es enviar un mensaje a nuestra lista de correo pública en <https://accounts.eclipse.org/mailling-list/ospo.zone>. También puede venir a debatir con nosotros en los eventos habituales sobre código abierto, participar en nuestros seminarios web mensuales OSPO OnRamp o ponerse en contacto con cualquier miembro, que amablemente le redirigirá a la persona adecuada.

10.2 Apéndice: Plantilla de cuadro de mando de actividades personalizado

La última versión de la plantilla del cuadro de mando de actividades personalizado está disponible en la sección resources del [Gitlab de la Good Governance Initiative](#) en OW2.

`../content/ggi_activities_metadata.json.md`

Manual de la Iniciativa para la Buena Gobernanza		LA EMPRESA DEL BUEN EJEMPLO		Cuadro de mando personalizado de actividades	
Objetivo/Actividad Cultura 1		Promover las mejores prácticas de desarrollo de código abierto		Última actualización 2024-03-04	
Descripción personalizada <i>Alcance de lo que hay que hacer</i> Breve descripción esencial... <ul style="list-style-type: none">Breves puntos destacados...			Evaluación de oportunidades <i>Por qué es relevante esta actividad</i> <ul style="list-style-type: none">Puntos clave de dolor...Oportunidades de progreso clave...		
Objetivos <i>Lo que pretendemos conseguir en esta iteración</i> <ul style="list-style-type: none">Objetivo 1...Objetivo 2...		Herramientas <i>Tecnologías, herramientas y productos utilizados en la actividad</i> <ul style="list-style-type: none">Recursos...		Notas operativas <i>Enfoque, método para progresar en la actividad</i> <ul style="list-style-type: none">Empieza con...	
Resultado clave <i>Cómo mediremos el éxito en esta iteración</i>			Progreso	Puntuación	Evaluación personal
1. Resultado clave 1 (mínimo un resultado clave)			xx%	0,9	Comentario personal
2. Resultado clave 2			xx%	0,5	Comentario personal
3. Resultado clave 3			xx%	0,5	Comentario personal
4. Resultado clave 4 (máximo cuatro resultados clave)			xx%	0,0	Comentario personal
				0,475	
Línea de tiempo <i>Fechas de inicio y fin, hitos</i> <ul style="list-style-type: none">Indicación de la fecha aquí		Esfuerzos <i>Presupuesto de tiempo y material</i> <ul style="list-style-type: none">Asignación de tiempo para los próximos tres mesesAsignación presupuestaria		Asignados <i>¿Quién participa? ¿Lleva la batuta?</i> <ul style="list-style-type: none">XX para preparar la presentación interna	
Asuntos <i>Dificultades, incertidumbres, bloqueos, puntos de atención, dependencias</i> <ul style="list-style-type: none">Preocupación 1...Preocupación 2...			Estado <i>Cómo va la actividad</i> Comentario personal sobre la salud de la actividad		
			Calificación general de progreso		XX%.
Notas					

Observaciones del foro de actividad de GitLab

https://gitlab.ow2.org/ggi/ggi/-/blob/main/handbook/content/52_activity_44.md

Copie/pegue aquí el contenido de la descripción de la actividad de <https://gitlab.ow2.org/ggi/ggi/>

Esto servirá de referencia para ayudar a desarrollar el Cuadro de Mando de Actividades Personalizado