Open Source Good Governance Handbook

Authors: OSPO Alliance & The GGI participants

Version: v1.1

Date: 2022-11-08

Contents

1 Introduction

This document introduces a methodology to implement professional management of open source software in an organisation. It addresses the need to use open source software properly and fairly, safeguard the company from technical, legal and IP threats, and maximise the advantages of open source. Wherever an organisation stands on these topics this document proposes guidance and ideas to move forward and make your journey a success.

1.1 Context

Most large end-users and systems integrators already use Free and Open-Source Software (FOSS) either in their information systems or product and service divisions. Open source compliance has become an ever-growing concern, and many large companies have established compliance officers. However, while sanitising a company's open-source production chain – which is what compliance is about – is fundamental, users *must* give back to communities and contribute to the sustainability of the open-source ecosystem. We see open source governance encompassing the whole ecosystem, engaging with local communities, nurturing a healthy relationship with open source software vendors and service specialists. This takes compliance to the next level, and this is what open source *good* governance is about.

This initiative goes beyond compliance and liability. It is about building awareness in communities of end-users (often software developers themselves) and systems integrators, and developing mutually beneficial relationships within the European FOSS ecosystem.

OSS Good Governance enables organisations of all types -- companies, small and large, city councils, universities, associations, etc. -- maximise the benefits derived from open source by helping them align people, processes, technology and strategy. And in this area, that of maximising the advantages of open source, especially in Europe, everyone is still learning and innovating, with nobody knowing where they actually stand regarding state of the art in the domain.

This initiative aims to help organisations achieve these goals with:

- A structured catalog of **activities**, a roadmap for the implementation of professional management of open source software.
- A **management tool** to define, monitor, report and communicate about progress.
- A **clear and practical path for improvement**, with small, affordable steps to mitigate risks, educate people, adapt processes, communicate inwards and outwards the organisation's realm.
- **Guidance** and a range of **curated references** about open-source licensing, best practices, training, and ecosystem engagement to leverage open-source awareness and culture, consolidate internal knowledge and extend leadership.

This guide has been developed with the following requirements in mind:

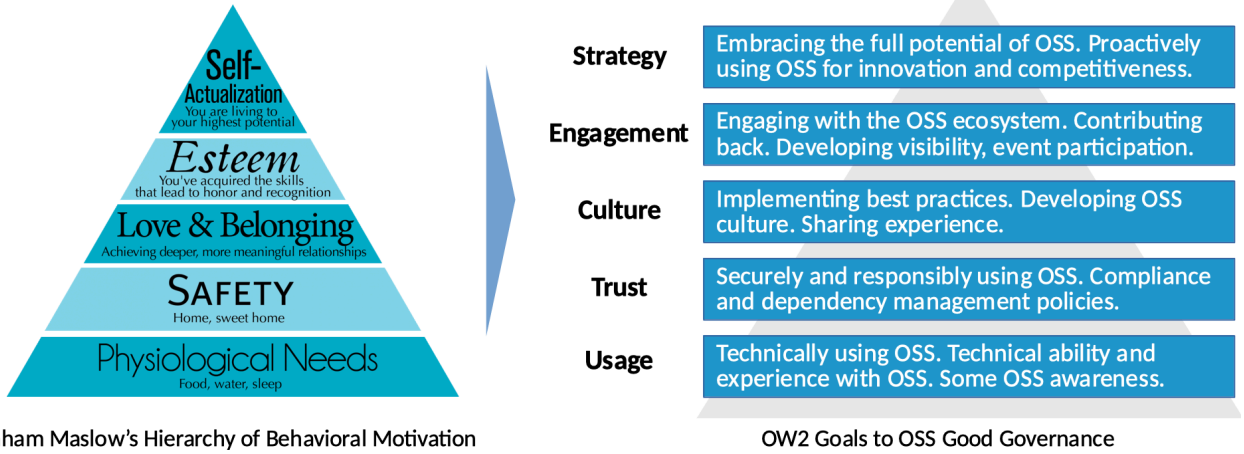
- Any type of organisation is covered: from SMEs to large companies and not-for-profit organisations, from local authorities (e.g. town councils) to large institutions (e.g. European or governmental institutions). The framework provides building blocks for a strategy and hints for its realisation, but *how* the activities are executed depends entirely on the program's context and is up to the program manager. It may prove helpful to look for consulting services and to exchange with peers.
- No assumption is made about the level of technical knowledge within the organisation or the domain of activity. For example, some organisations will need to set up a complete training curriculum, while others might simply propose ad-hoc material to the teams.

Some activities will not be relevant to all situations, but the whole framework still provides a comprehensive roadmap and paves the way for tailored strategies.

1.2 About the Good Governance Initiative

At OW2, an initiative is a joint effort to address a market need. The Good Governance Initiative proposes a methodological framework to implement professional management of open source software within organisations.

The Good Governance initiative is based on a comprehensive model inspired by the popular Abraham Maslow's hierarchy of human needs and motivations, as illustrated by the picture below.



Abraham Maslow's Hierarchy of Behavioral Motivation

OW2 Goals to OSS Good Governance

Through ideas, guidelines and activities the Good Governance initiative provides a blueprint for the implementation of organisational entities tasked with professional management of open source software, what is also called OSPO (for Open Source Program Offices). The methodology is also a management system to define priorities, and monitor and share progress.

As they implement the OSS Good Governance methodology, organisations will enhance their skills in a number of directions, including:

- **using** open source software properly and safely within the company to improve software reuse and maintainability and software development velocity;
- **mitigating** the legal and technical risks associated with external code and collaboration;
- **identifying** required training for teams, from developers to team leaders and managers, so everybody shares the same vision;
- **prioritizing** goals and activities, to develop an efficient open source strategy;
- **communicating** efficiently within the company and to the external world to make the most off the open source strategy;
- **improving** the organisation's competitiveness and attractiveness for top open source talents.

1.3 About the OSPO Alliance

The **OSPO Alliance** was launched by a coalition of leading European open source non-profit organizations, including OW2, Eclipse Foundation, OpenForum Europe, and Foundation for Public Code, with a mission to grow awareness for open source in Europe and globally and to promote the structured and professional management of open source by companies and administrations.

While the Good Governance initiative is focused on developing a management methodology, the OSPO Alliance has the broader goal to help companies, particularly in non-technology sectors, and public institutions discover and understand open source, start benefiting from it across their activities and grow to host their own OSPOs.

The OSPO Alliance has established the **OSPO Alliance** website hosted at https://ospo-alliance.org. The OSPO Alliance serves the community with a safe place to discuss and exchange on the topics of OSPOs, and provides a repository for a comprehensive set of resources for corporations, public institutions, and research and academic organizations. The OSPO Alliance connects with OSPOs across Europe and the world as well as with supportive community organizations. It encourages best practices and fosters contribution to the sustainability of the open source ecosystem. Check out the OSPO Alliance website for a quick overview of complementary frameworks of IT management best practices.

The OSPO Alliance website is also the place where we collect feedback about the initiative and its content (e.g. activities, body of knowledge) from the community at large.

1.4 Translations

There is an ongoing community work to translate the GGI Handbook in various languages. As progress evolves rapidly, we recommend to check out our official website for a complete list of available translations.

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

The GGI handbook is translated using Weblate, an open source project and platform that offers free hosting for open source projects. We want to thank them deeply, as well as all our translation contributors. You are amazing.

1.5 Contributors

The following great people have contributed to the Good Governance Initiative:

- Frédéric Aatz (Microsoft France)
- Boris Baldassari (Castalia Solutions, Eclipse Foundation)
- Philippe Bareille (Ville de Paris)
- 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)
- Hervé Pacault (Orange)
- Stefano Pampaloni (RIOS)
- Christian Paterson (OpenUp)
- Simon Phipps (Meshed Insights)
- Silvério Santos (Orange Business Services)
- Cédric Thomas (OW2)
- Nicolas Toussaint (Orange Business Services)

1.6 Licence

2 Organisation

2.1 Terminology

The OSS Good Governance methodology blueprint is structured around four key concepts: Goals, Canonical Activities, Customised Activity Scorecards and Iteration.

- **Goals**: A Goal is a set of activities associated with a common area of concern, there are five Goals: Usage Goal, Trust Goal, Culture Goal, Engagement Goal and Strategy Goal. Goals can be achieved independently, in parallel, and iteratively refined through Activities.
- **Canonical Activities**: within a Goal, an Activity addresses a single concern or topic of development -- such as Managing legal compliance -- that can be used as an incremental step towards the program's objectives. The complete set of Activities as defined by the GGI are called the Canonical Activities.
- **Customised Activity Scorecard (CAS)**: To implement GGI in a given organisation, the Canonical Activities must be adapted to the specifics of the context, thus building a set of Customised Activity Scorecards. The Customised Activity Scorecard describes how the activity will be implemented in the context of the organisation and how progress will be monitored.
- **Iteration**: The OSS Good Governance is a management system and as such requires periodic assessment, review and revision. Think of the accounting system in an organisation, it's an on-going process with at least one annual check point, the balance sheet; in the same way, the OSS Good Governance process requires at least an annual review, however reviews can be partial or more frequent depending on the Activities.

2.2 Goals

The Canonical Activities defined by the GGI are organised in Goals. Each Goal addresses a specific area of progress within the process. From Usage to Strategy, Goals cover issues related to all stakeholders, from development teams up to C-level management.

- **Usage** Goal: This Goal covers the basic steps in using open source software. Activities related to the Usage Goal cover the first steps through an open source program, by identifying how efficiently open source is used and what it brings to the organisation. It includes training and knowledge management, producing inventories of existing open source already used in-house, and presents some open source concepts that can be used throughout the process.
- **Trust** Goal: This Goal is concerned about using open source securely. The Trust Goal deals with legal compliance, dependency and vulnerability management and generally aims to build confidence in how the organisation uses and manages open source.
- **Culture** Goal: The cultural objective includes activities aimed at making teams comfortable with open source, individually participating in collaborative activities, understanding and implementing open source best practices. This objective fosters a sense of belonging to the open source community among individuals.
- **Engagement** Goal: This goal is to engage with the open source ecosystem at the corporate level. Human and financial resources are budgeted to contribute back to open source projects. Here, the organisation asserts that it is a responsible open source citizen and acknowledges its responsibility to ensure the sustainability of the open source ecosystem.
- **Strategy** goal: This Goal is about making open source visible and acceptable at the highest levels of corporate management. It is about recognising that open source is a strategic enabler of digital sovereignty, process innovation and, in general, a source of attractiveness and goodwill.

2.3 Canonical Activities

The Canonical Activities are at the centre of the GGI blueprint. In its initial version, the GGI Methodology provides five Canonical Activities per goal, 25 in total. Canonical Activities are described using the following predefined sections:

- *Description*: a summary of the topic that the activity addresses and the steps to completion.
- *Opportunity Assessment*: describes why and when it is relevant to undertake this activity.

- *Progress Assessment*: describes how to measure progress on the activity and to assess its success.
- *Tools*: a list of technologies or tools that can help achieve this activity.
- *Recommendations*: hints and best practices collected from GGI participants.
- *Resources*: links and references to read more about the topic covered by the activity.

Description

This section provides a high-level description of the Activity, a summary of the topic to set the purpose of the activity in the context of the open source approach within a goal.

Opportunity Assessment

To help structure an iterative approach, each activity has an ”Opportunity Assessment” section, with one or more questions attached to it. The opportunity assessment focuses on why it is relevant to undertake this activity, what needs it addresses. Assessing the opportunity will help define what are the efforts expected, resources needed, and help evaluate costs and expected ROI.

Progress Assessment

This step focuses on defining objectives, KPIs, and on providing *verification points* that help evaluate progress in the Activity. The verification points are suggested, they can help define a roadmap for the Good Governance process, its priorities and how progress will be measured.

Tools

Here are listed Tools that can help in delivering the activity or instrument a specific step of the activities. Tools are not a mandatory recommendation, nor pretend to be exhaustive, but are suggestions or categories to be elaborated upon based on existing context.

Recommendations

This section is regularly updated with feedbacks from users and all sorts of recommendations that can help manage the Activity.

Resources

Resources are proposed to feed the approach with background studies, reference documents, events or online content to enrich and develop the related approach on the activity. Resources are not exhaustive, they are starting points or suggestions to expand on the semantics of the activity according to one’s own context.

2.4 Customised Activity Scorecards

Customised Activity Scorecards (CAS) are slightly more detailed than Canonical Activities. A CAS includes details specific to the organisation implementing GGI. Using the CAS is described in the Methodology section.

3 Methodology

Implementing the OSS Good Governance methodology is ultimately a consequential and impactful initiative. It involves several categories of company people, services, and processes, from everyday practices to HR management and from developers to C-level executives. There really is no silver bullet mechanism to implement open source good governance. Different types of organisation and company cultures and situations will demand different approaches to open source governance. For each organisation, there will be different constraints and expectations, leading to different paths and ways of managing the programme.

With this in mind the Good Governance Initiative provides a generic blueprint of activities that can be tailored to an organisation's own domain, culture and requirements. While the blueprint claims to be comprehensive, the methodology can be implemented progressively. It is possible to bootstrap the program by simply selecting the most relevant Goals and Activities in one's specific context. The idea is to build a first-draft roadmap to help set up the local initiative.

Besides this framework, we also highly recommend getting in touch with peers through an established network like the European OSPO Alliance initiative, or other like-minded initiatives from the TODO group or OSPO++. What is important is to be able to exchange with people running a similar initiative, and share the issues encountered and the solutions that exist.

3.1 Setting the Stage

Given the ambition of the good governance methodology and its potentially broad impact, it is important to communicate with a variety of people within an organisation. It would be appropriate to onboard them to establish an initial set of realistic expectations and requirements to get off to a good start, attract interest and support. A good idea is to publish the Customised Activity Scorecards on the organisation's collaborative platform so they can be used to communicate with stakeholders. Some hints:

- Identify key stakeholders, make them agree on a set of primary objectives. Engage them in the success of the initiative as a part of their own agenda.
- Get initial buy-in, agree on the steps and pace, and set up regular checks to inform them of progress.
- Make sure they understand the benefits of what can be achieved and what it involves: expected improvement should be clear and outcome visible.
- Establish a first diagnostic or state of the art of open source in the candidate organisation. Outcome: a document describing what this program will achieve, where the organisation stands and where it aims to go.

3.2 Workflow

As modern software practitioners, we like agile-like methods that define small and safe increments, since it is good practice to reassess the situation regularly and to provide meaningful minimum intermediate results.

In the context of a live OSPO program this is highly relevant, as many side aspects will change over time, from the organisation's strategy and response to open source to people's availability and engagement. Periodic reassessment and iteration also allows for adaptation to ongoing programme acceptance, better tracking of current trends and opportunities, and small incremental benefits to stakeholders and the organisation as a whole.

Ideally, the methodology could be implemented in five phases as follows:

1. **Discovery** Understanding key concepts, taking ownership of the methodology, aligning goals expectations.
2. **Customisation** Adapting Activity description and opportunity assessment to organisation specifics.
3. **Prioritisation** Identifying objectives and key results, tasks and tools, scheduling milestones and drafting timeline.
4. **Activation** Finalising Scorecard, budget, assignments, document tasks on issue manager.
5. **Iteration** Assessing and scoring results, highlighting issues, improving, adjusting. Iterate every quarter or semester.

Preparing for the first program iteration:

- Identify a first set of tasks to work on, and prioritise them according to the needs (gaps to the desired state) and the timeline. Outcome: a list of tasks to work on during the iteration.
- Define a set of requirements and areas of improvement, communicate it to the stakeholders and end-users, and get their approval or commitment.
- Fill the scorecards for progress tracking. A template scorecard can be downloaded from the GGI repository.

At the end of each iteration, do a retrospective and prepare for the next iteration:

- Communicate about the latest improvements.
- Assess where you are, if the targeted tasks have been completed, refine the roadmap accordingly.
- Check the remaining pain points or issues, ask for support from other actors or services if needed.
- Re-prioritise tasks according to the updated context.
- Define a new subset of tasks to execute.

3.3 Manual setup: using Customised Activity Scorecards

A Customised Activity Scorecard is a form describing a Canonical Activity customised to the specifics of an organisation. Put together, the deck of Customised Activity Scorecards provides the roadmap for managing open source software.

`Please note, from early experience with the methodology, it takes up to one hour to adapt a Canonical Activity into an organisation's specific Customised Score Card.`

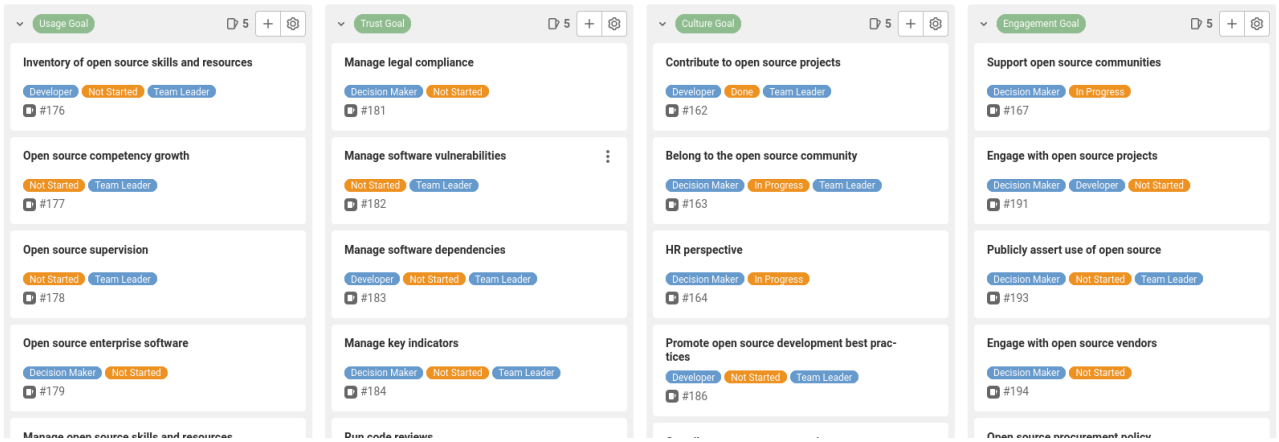
The Customised Activity Scorecard contains the following sections:

- **Title Disambiguation** First of all take a few minutes to develop an understanding of what the Activity might be about and its relevance, how it can fit in your overall OSS management journey.
- **Customised Description** Adapt the Activity to the specifics of the organisation, scoping. Define the scope of the Activity, the particular use case you will address.
- **Opportunity Assessment** Explain why it is relevant to undertake this activity, what needs it addresses. What are our pain points? What are the opportunities for progressing? What can be gained?
- **Objectives** Define a couple of crucial objectives for the Activity. Pain points to be fixed, progress opportunities, wishes. Identify key tasks. What we aim to achieve in this iteration.
- **Tools** Technologies, tools and products used in the Activity.
- **Operational Notes** Indications on approach, method, strategy to progress in this Activity.
- **Key Results** Define measurable, verifiable expected results. Choose results indicating progress with regard to the Objectives. Indicate KPIs here.
- **Progress and Score** Progress is, in %, the completion rate of the result; Score is the personal success rating.
- **Personal Assessment** For each result you can add a brief explanation and explain your personal satisfaction rate expressed in the Score.
- **Timeline** Indicate Start-End dates, Phasing tasks, Critical steps, Milestones.
- **Efforts** Evaluate requested time and material resources, internal and third-party. What are the efforts expected? How much will it cost? What resources do we need?
- **Assignees** Say who participates. Assign tasks or Activity leadership and responsibilities.
- **Issues** Identify key issues, foreseen difficulties, risks, roadblocks, uncertainties, points of attention, critical dependencies.
- **Status** Write here a synthetic assessment of how the Activity is doing: healthy? Delayed? Etc.
- **Overall Progress Rating** Your own high-level, management-oriented, synthetic Activity progress evaluation.

3.4 Automatic setup: using the GGI Deployment feature

Starting with the Handbook 1.1 version, the GGI proposes My GGI Board, an automated tooling to deploy your own instance of the GGI as a GitLab project. The install process takes less than

10mn to set up, is fully documented, and provides a simple and reliable way to customise the activities, follow their execution as you make progress, and communicate the results to your stakeholders. A live example of the deployment can be seen in the initative's GitLab, with the auto-generated website available on its GitLab pages.



Here is a standard workflow to use the deployment feature:

1. Fork the My GGI Board to your own GitLab instance or project, and set it up following the instructions in the project's README: https://gitlab.ow2.org/ggi/my-ggi-board. This will:

- Create all the activities as issues in the project.
- Create a nice board to help you visualise and manage the activities.
- Create a static website, served on your GitLab instance pages, with the information extracted from the activities.
- Update the project's description with the proper links to the board of activities and your static website.

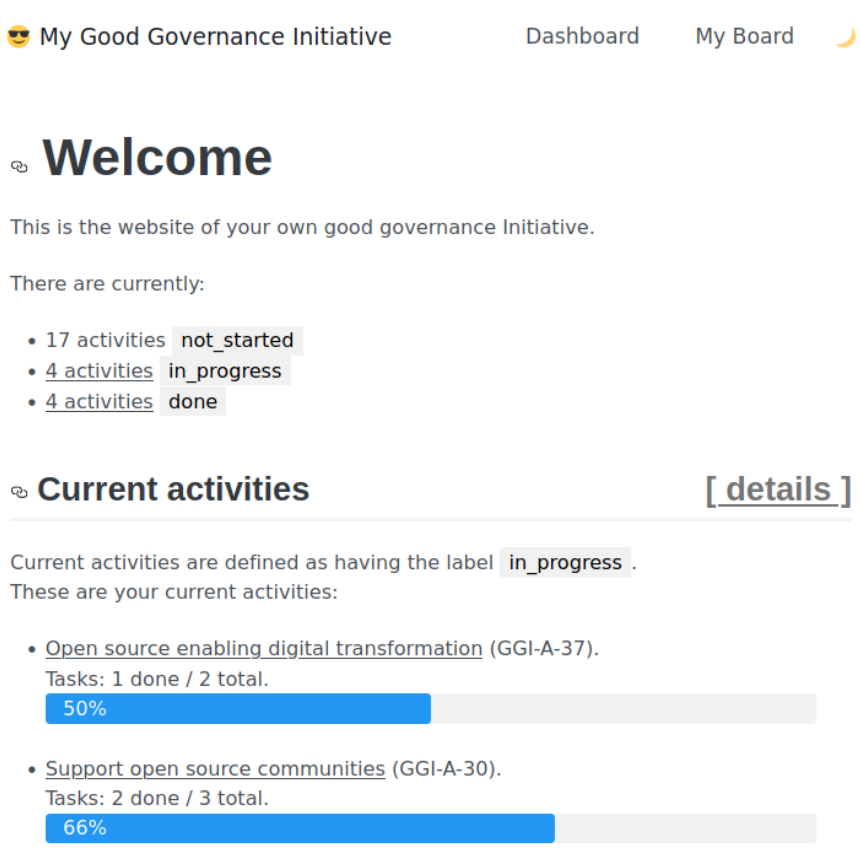
2. From there, you can start looking at the different activities and filling in the scorecard section.

- The scorecard section is the electronic (and simplified) equivalent of the above-mentioned ODT scorecards. They are used to adapt the activity to your context, by listing the local resources, risks and opportunities, and defining custom objectives required to complete the activity.
- If some activity does not apply to your context, then simply mark it as 'Not Selected', or close it.
- This is a quite time-consuming process, but highly needed as it will help you, step-by-step, to define your own roadmap and plan.

3. When activities have been defined, you can start implementing your own OSPO. Select a few activities that you think are relevant to start with, and change their progress label from 'Not Started' to 'In Progress'. You can use GitLab features to help you organise the work (comments, assignees, etc.) or any other tool. It's easy to link to the activities, and there are plenty of great integrations available.

4. On a regular basis (weekly, monthly, depending on your timetable), assess and review the current activities and when they are completed, change the label from 'In Progress' to 'Done'. Select a few other and start again at step 3 until they are all completed.

The website proposes a quick overview of the current and past activities, and extracts the scorecard section of issues to display only the locally relevant information. When changes happen in the issues (activities) these are automatically updated in the generated website. Please note that the CI pipelines for the automatic generation of the website are automatically executed nightly, but you can easily launch them from the GitLab project's CI/CD section. The following picture shows the auto-generated website interface.



You can ask questions or get support for the deployment feature at our GitLab homepage, and we welcome feedback.

GGI Deploy homepage: https://gitlab.ow2.org/ggi/my-ggi-board

3.5 Enjoy

Communicate on your success and enjoy the peace of mind of a state of the art open source strategy!

The OSS Good Governance is a method to deploy a continuous improvement program, and as such it never ends. Nevertheless, it's important to highlight intermediate steps and appreciate the changes it yields, to make progress visible and share the results.

- Communicate with stakeholders and end-users to let them know the advantages and benefits that the initiative's effort brings.
- Foster sustainability of the program. Ensure that the best practices and lessons learned from the program are always applied and updated.
- Share your experience with your peers: provide feedback to the GGI working group and within your OSPO community of adoption, and share your approach.

4 Usage goal activities

4.1 Inventory of open source skills and resources

Activity ID: GGI-A-17.

Description

At any stage, from a management perspective, it is useful to have a mapping, an inventory of open source resources, assets, usage and their status, as well as potential needs and available solutions. It also includes assessing the required effort and skills to fill the gap.

This activity aims to take a snapshot of the open source situation within the organisation and on the market and evaluate the bridge between them.

- Inventory of OSS usage in the software development chain as well as in the software products and components used in production.
- Identify open source technologies (solutions, frameworks, innovative features) that could fit your needs and help improve your process.

Not included

- Identify and qualify related OSS ecosystems and communities. (Culture Goal)
- Identify dependencies on OSS libraries and components. (Trust Goal)
- Identify the technical (e.g. languages, frameworks..) and soft (e.g. collaboration, communication) skills needed. (belongs to next Activities: OSS competency growth and Open source software development skills)

Opportunity Assessment

An inventory of available open source resources that will help optimise investment and prioritise skills development.

This activity creates the conditions for improving development productivity given the efficiency and popularity of OSS components, development principles and tools, particularly in the development of modern applications and infrastructures.

- This may require simplifying the portfolio of OSS resources.
- This may require retraining personnel.
- This enables the identification of needs and feeds your IT roadmap.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is a workable list of OSS resources ”We use”, ”We integrate”, ”We produce”, ”We host”, and the related Skills
- ☐ We are on a path to improve efficiency by using state of the art methods and tools.
- ☐ We have identified OSS resources unaccounted for until now (that may have been creeping in, and do we have elements to define policy in this domain?)
- ☐ We request new projects to endorse or reuse existing OSS resources. (Culture Goal?)
- ☐ We have a reasonably safe perception and understanding of the scope of OSS usage in our organisation.

Tools

There are many different ways to establish such inventory. One way would be to classify OSS resources into four categories:

- OSS we use: software we use either in production or in development
- OSS we integrate: for example, OSS libraries we integrate into a custom-made application
- OSS we produce: for example, a library we have published on GitHub or an OSS project we develop or regularly contribute to.
- OSS we host: OSS we run to offer an in-house service such as a CRM, GitLab, nexus, etc.

An example table would look like the following:

We use	We integrate	We produce	We host	Skills
Firefox, OpenOffice, Postgresql	Library slf4j	Library YY on GH	GitLab, Nexus	Java, Python

The same identification should apply to skills

- Skills & experiences available through the existing teams
- Skills & experiences that could be developed or acquired internally (training, coaching, experiment)
- Skills & experiences that need to be sought out on the market or through partnership / contracting

Recommendations

- Keep things simple.
- It's a relatively high-level exercise, not a detailed inventory for the accounting department.
- While this activity is a good starting point, you do not need to have it 100% completed before launching other activities.
- Handle issues, resources and skills related to **software development** in Activity #42.
- The inventory should cover all IT categories: operating systems, middlewares, DBMS, system administration, development and testing tools, etc.
- Start identifying related communities: it's easier to get support and feedback from the project when they already know you.

Resources

- An excellent course on Free (/Libre), and Open Source Software (FOSS), by Professor Dirk Riehle.

4.2 Open source competency growth

Activity ID: GGI-A-18.

Description

This activity is about planning and initiating technical abilities and early experience with OSS once an inventory has been conducted (#17). It is also the opportunity to start establishing a basic, lightweight skills development roadmap.

- Identify what the required skills and training are.
- Set up a pilot project to kick start the approach, learn from doing, establish a first achievement milestone.
- Capitalise on lessons learned and build a body of knowledge.
- Start identifying and documenting the next steps for broader adoption.
- Elaborate a strategy over the next few months or a year to engage management and financial support.

The scope of the activity:

- Linux, Apache, Debian, administration skills.
- Open source databases MariaDB, MySQL, PostgreSQL, etc.
- Open source virtualisation and cloud technologies.
- LAMP stack and its alternatives.

Opportunity Assessment

Like any IT technology, and probably even more, open source brings innovation. Open source grows fast and changes quickly. It requires organisations to keep up to date.

This activity helps identify areas where training could help people become more efficient and feel more secure using open source. It helps make employee development decisions. Seeding basic open source skills allows evaluating the opportunity to:

- Extend IT solutions with existing market technologies developed by the ecosystem.
- Develop new ways of collaboration inside and outside the organisation.
- Acquire competencies in new and innovative technologies.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ A skills matrix is developed.
- ☐ The scope of OSS technologies used is proactively defined, i.e. avoiding uncontrolled use of OSS technologies.
- ☐ A satisfactory level of expertise is acquired for these technologies.
- ☐ Teams have received an ”open source Basics” training to get started.

Tools

A key tool here is called Activity (or Competency) Matrix (or Mapping).

This activity can be conducted by:

- using online tutorials (many free of charge on the Internet),
- participating in developers conferences,
- receive vendor training, etc.

Recommendations

- Using and developing open source components in a safe and efficient manner requires an open, collaborative mindset that needs to be recognised and propagated both from the top (management) and the bottom (developers).
- Make sure that the approach is actively supported and promoted by the management. Nothing will happen if there is no commitment from the hierarchy.
- Involve people (developers, stakeholders) in the process: organise round tables and listen to ideas.
- Allow time and resources for people to discover, try and play with these new concepts. If possible, make it fun -- gamification and rewards are good incentives.

A pilot project with the following steps could serve as a catalyst:

- Identify the technology or framework to start with.
- Find online training, tutorial, and sample code to experiment.
- Build a prototype of the end solution.
- Identify some experts to challenge and coach on implementation.

Resources

- What is a Competency Matrix: a quick introductory read.
- How to Make a Skills Matrix for your Team: a template with commentaries.
- MOOC on Free (libre) culture (French Only): this is a 6 part course on the free culture, introduction to Copyrights, Intellectual Property, open source licensing

4.3 Open source supervision

Activity ID: GGI-A-19.

Description

This activity is about controlling the use of open source and ensuring open source software is proactively managed. This concerns several perspectives, be it to use OSS tools and business solutions, or to include OSS as components in own developments or modify a version of a software adapting it to its own needs, etc. It is also about identifying areas where open source has become a (sometimes covert) de facto solution and assessing its suitability.

It may be necessary to clarify the following:

- Is the required functionality provided?
- Is there additional functionality provided that is not needed but is increasing complexity in the BUILD and RUN phases?
- What does the licence require, what are the legal constraints?
- How much does the decision make your organisation supplier-independent?
- Does a support option, ready for your business needs, exist, and how much does it cost?
- TCO (Total Cost of Ownership).
- Does the management know about open source's advantages, e.g. beyond "saving licence cost"? Being comfortable with open source helps get the maximum benefit from working with project communities and vendors.
- See if it makes sense to share development costs by giving one's developments to the community and all its implications, like licence compliance.
- Check for availability of community support or professional support.

Opportunity Assessment

Defining a decision process specifically directed at open source is a way to maximise its benefits.

- It avoids the uncontrolled creeping usage and hidden costs of OSS technologies.
- It leads to informed and OSS-aware strategic and organisational decisions.

Costs: the activity may challenge and reconsider sub-optimal de facto use of open source as inefficient, risky, etc.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ OSS has become a comfortable option when selecting OSS is not seen as an exception or a dangerous choice.
- ☐ OSS has become a "mainstream" option.
- ☐ Key players are sufficiently convinced the open source solution has strategic advantages worth investing in.
- ☐ It can be demonstrated that the TCO of the solution based on open source gives your organisation a higher value than the alternative.
- ☐ There is an evaluation of how supplier independence saves money or potentially can save money in the future.
- ☐ There is an evaluation that solution independence reduces risks to be too expensive to change the solution (no closed data formats possible).

Tools

At this stage, we cannot think of any tool relevant or concerned by this activity.

Recommendations

- Proactively managing the use of open-source requires basic levels of awareness and understanding of open source fundamentals because they should be considered in any OSS decision.
- Compare the needed functionality instead of looking for an alternative for a known closed source solution.
- Make sure to have support and further development.
- Regard the effects of the solution's licence on your organisation.
- Convince all key players about the value of the advantages of open source, beyond "saving licence cost".
- Be honest, do not exaggerate the open source solution's effect.
- In the decision-making process it is equally important to assess different open source solutions in order to avoid disappointment through wrong expectations, to make it clear what the organisation is required to do and all the advantages the openness of the solutions brings. This must be identified so the organisation can assess it for its own context.

Resources

- Top 5 Benefits of Open Source: Sponsored blog, but still interesting, quick read.
- Weighing The Hidden Costs Of Open Source: an IBM-sponsored look at OSS support costs.

4.4 Open source enterprise software

Activity ID: GGI-A-20.

Description

This activity is about proactively selecting OSS solutions, either vendor or community-supported, in business-oriented areas. It may also cover defining preference policies for the selection of open source business application software.

While open source software is most often used by IT professionals -- operating system, middleware, DBMS, system administration, development tools -- it has yet to be recognized in areas where business professionals are the primary users.

The activity concerns areas such as: Office suites, Collaboration environments, User management, Workflow management, Customer relationship management, Email, e-Commerce, etc.

Opportunity Assessment

As open source becomes mainstream it reaches out well beyond operating systems and development tools, it increasingly finds its way into the upper layers of the information systems, well into the business applications. It is relevant to identify what OSS applications are successfully used to meet the needs of the organisation and how they can become an organisation's cost-saving preferred choice.

The activity may bring some retraining and migration costs.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is a list of recommended OSS solutions to address pending needs in business applications.
- ☐ A preference policy for the selection of open source business application software is drafted.
- ☐ Proprietary business applications in use are being evaluated against OSS equivalents.
- ☐ Procurement process and calls for proposals specify open source preference (if legally doable).

Tools

Tools to map out software and business applications?

At this stage, we cannot think of any tool relevant or concerned by this Activity.

Recommendations

- Talk to colleagues, learn from what other companies comparable to yours do.
- Visit local industry events to find out about OSS solutions and professional support.
- Try out community editions and community support first before committing to paid support plans.

Resources

- What is enterprise open source?: a quick read about enterprise-ready open source.
- 101 Open Source Apps to Help your Business Thrive: An indicative list of business-oriented open source solutions.

4.5 Manage open source skills and resources

Activity ID: GGI-A-42.

Description

This activity is focused on **software development** skills and resources. It includes the technologies and specific development skills of developers, as well as the overall development processes, methods and tools.

A vast amount of documentation, forums and discussions stemming from the ecosystem, and public resources is available for open source technologies. In order to fully benefit from their open source approach, one has to establish a roadmap of its current assets and desired targets to set up a consistent program for development skills, methods and tools within the teams.

Domains of application One needs to establish the domains where the program will be applied, and how it will improve the quality and efficiency of code and practices. As an example, the program won't have the same benefits if there is only a single developer working on open source components, or if the whole development life cycle is optimised to include open source best practices.

One needs to define the scope to be embraced for open source development: technical components, applications, modernising or creating new development. Examples of development practices that can benefit from open source are:

- Cloud administration.
- Cloud-native applications, how to innovate with these technologies.
- DevOps, Continuous Integration / Continuous Delivery.

Categories

- Skills and resources required to develop open source software: IP, licensing, practices.
- Skills and resources required to develop software using open source components, languages, technologies.
- Skills and resources required to use open source methods and processes.

Opportunity Assessment

Open source tools are increasingly popular among developers. This Activity addresses the need to avoid the proliferation of heterogeneous tools within a development team. It helps define a policy in this domain. It helps optimise training and experience building. A skills inventory is used for recruitment, training and succession planning in case a key employee leaves the company.

We would need a methodology to map open source software development skills.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is a description of the open source production chain (the "software supply chain"),
- ☐ There is a plan (or a wish list) for the rationalisation of development resources,
- ☐ There is a skills inventory summarising current developers' skills, education, and experience,
- ☐ There is a training wish list and program dealing with skills gaps,
- ☐ There is a list of missing open source development best practices and a plan to apply them.

Recommendations

- Start simple, grow the analysis and roadmap steadily.
- When recruiting, set a strong emphasis on open source skills and experience. It's always easier when people already have an open source DNA than training and coaching people.
- Check training programs from software vendors and open source schools.

Resources

More information:

- An introduction to what is a Skills Inventory? from Robert Tanner.
- An article about open source skills: 5 Open Source Skills to Up Your Game and Your Resume

This activity can include technical resources and skills such as:

- **Popular languages** (such as Java, PHP, Perl, Python).
- **Open source frameworks** (Spring, AngularJS, Symfony) and testing tools.
- Agile, DevOps and open source **development methods and best practices**.

Associated activities:

- GGI-A-28 HR perspective

5 Trust goal activities

5.1 Manage legal compliance

Activity ID: GGI-A-21.

Description

Organisations need to implement a legal compliance process to secure their usage and participation in open source projects.

Mature and professional management of legal compliance, in the organisation and across the supply chain, is about:

- Performing a thorough analysis of the intellectual property that includes licence identification and compatibility checking.
- Ensuring the organisation can safely use, integrate, modify and redistribute open source components as part of its products or services.
- Providing employees and contractors with a transparent process about how to create and contribute to open source software.

Software Composition Analysis (SCA): A significant part of legal and IP issues result from the usage of components released under licences that are either incompatible between them or incompatible with the way the organisation wants to use and redistribute the components. SCA is the first step in sorting out those issues as ”you need to know the problem to fix it”. The process is to identify all the components involved in a project in a Bill of Material document, including build and test dependencies.

Licence checking: A licence checking process uses a tool to automatically analyse the code base and identify licences and copyrights within. If executed regularly and ideally integrated into continuous build and integration chains, this allows catching IP issues early.

Opportunity Assessment

With the ever-growing use of OSS in an organisation’s information systems, it is essential to assess and manage potential legal exposure.

However, checking licences and copyrights can be tricky and costly. Developers need to be able to check IP and legal questions quickly. Having a team and a corporate officer dedicated to IP and legal questions ensures proactive and consistent management of legal questions, helps secure open source components’ usage and contributions and provides a clear strategic vision.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is an easy-to-use Licence checking process available for projects.
- ☐ There is an easy-to-use IP checking process available for projects.
- ☐ There is a team or person responsible for legal compliance within the organisation.
- ☐ Regular audits to assess legal compliance are scheduled.

Other ways to set up verification points:

- ☐ There is an easy-to-use licence checking process.
- ☐ There is an easy-to-use legal/IP team as in activity #13.
- ☐ All projects provide the required information for people to use and contribute to the project.
- ☐ There is a contact in the team for questions related to IP and licencing.
- ☐ There is a corporate officer dedicated to IP and licencing.
- ☐ There is a dedicated team for questions related to IP and licencing.

Tools

- ScanCode
- Fossology
- SW360

- Fossa
- OSS Review Toolkit

Recommendations

- Inform people about the risks associated with licensing in conflict with business goals.
- Propose an easy solution for projects to set up licence checking on their codebase.
- Communicate on its importance and help projects to add it to their CI systems.
- Provide a template or official guidelines for project structure.
- Set up automated checks to make sure that all projects comply with the guidelines.
- Consider conducting an internal audit to identify licences of the company infrastructure.
- Provide basic IP and licensing training for at least one person per team.
- Provide complete IP and licencing training for the officer.
- Set up a process to escalate IP and licencing issues to the officer.

Remember that compliance is not just about legal; it's also about IP. So here are a few questions to help understand the consequences of legal compliance:

- If I distribute an open source component and do not respect the licence conditions, I infringe the licence --> legal implications.
- If I use an open source component within a project that I wish to distribute/publish, that licence may oblige visibility on elements of code that I do not want to make open source --> Confidentiality impact for my company's tactical advantage and with 3rd parties (legal implications).
- It is an open discussion about whether using an open source licence for a project I want to publish grants relevant IP --> IP implications.
- If I make a project open source *before* any patent process, that *probably* excludes the creation of patents concerning the project --> IP implications.
- If I make a project open source *after* any patent process, that *probably* allows the creation of (defensive) patents concerning that project --> IP potential.
- In complex projects that bring in many components with many dependencies, the multitude of open source licences may exhibit incompatibilities between licences --> legal implications (cf. Issue #23).

Resources

- There is an extensive list of tools on the Existing OSS compliance group page.
- Recommended Open Source Compliance Practices for the enterprise. A book by Ibrahim Haddad, from the Linux Foundation, about open source compliance practices for the enterprise. OpenChain Project

5.2 Manage software vulnerabilities

Activity ID: GGI-A-22.

Description

One's code is as secure as its least secure part. Recent cases (e.g. heartbleed[1], equifax[2]) have demonstrated the importance of checking vulnerabilities in parts of the code that are not directly developed by the entity. Consequences of exposures range from data leaks (with tremendous reputational impact) to ransomware attacks and business-threatening unavailability of services.

Open source software is known to have better vulnerability management than proprietary software, mainly because:

- More eyes are looking to find and fix problems on open code and processes.
- Open source projects fix vulnerabilities and release patches and new versions a lot faster.

[1] https://fr.wikipedia.org/wiki/Heartbleed

[2] https://arstechnica.com/information-technology/2017/09/massive-equifax-breach-caused-by-failure-to-patch-two-month-old-bug/

For example, a study by WhiteSource on proprietary software showed that 95% of the vulnerabilities found in their open source components had already released a fix at the time of the analysis. The issue, therefore, is to **better manage vulnerabilities both in the codebase and its dependencies**, no matter if they are closed or open source.

In order to mitigate these risks, one has to set up an assessment program of its software assets and a vulnerability-checking process executed regularly. Implement tools that alert impacted teams, manage known vulnerabilities, and prevent threats from software dependencies.

Opportunity Assessment

Any company that uses software has to watch its vulnerabilities in:

- its infrastructure (e.g. Cloud infrastructure, network infrastructure, data stores),
- its business applications (HR, CRM tools, internal and customers-related data management),
- its in-house code: e.g. the company's website, internal development projects, etc.,
- and all direct and indirect software and services dependencies.

The ROI of vulnerabilities is little known until something bad happens. One has to consider the consequences of a major data breach or unavailability of services to estimate the true cost of vulnerabilities.

Similarly, a culture of secrecy and hiding for security-related issues inside the company has to be avoided at all costs. Instead, information about the state of vulnerability needs to be shared and discussed to find the best answers from the right people, from developers to c-level executives.

The benefits of preventing cyber-attacks by carefully managing software vulnerabilities are manyfold:

- Avoid reputational risks,
- Avoid exploitation loss (DDoS, Ransomware, Time to rebuild an alternative IT system after an attack),
- Comply with data protection regulations.

Managing OSS software vulnerabilities is just a part of the larger cybersecurity process that globally addresses the security of the systems and services in the organisation.

Progress Assessment

There should be a dedicated person or team to monitor vulnerabilities and easy-to-use processes for developers to rely on. Vulnerabilities assessment is a standard part of the continuous integration process, and people are able to monitor the current state of risk in a dedicated dashboard.

The following **verification points** demonstrate progress in this Activity:

- ☐ Activity is covered when all in-house software and services are assessed and monitored for known vulnerabilities.
- ☐ Activity is covered when a dedicated tool and process is implemented in the software production chain to prevent the introduction of issues in the daily development routines.
- ☐ A person or team is responsible for evaluating CVE/vulnerability risk against exposure.
- ☐ A person or team is responsible for dispatching CVE/vulnerability to concerned people (SysOps, DevOps, developers, etc.).

Tools

- GitHub tools
  - GitHub provides guidelines and tools to secure code hosted on the platform. See GitHub docs for more information.
  - GitHub provides Dependabot to identify vulnerabilities in dependencies automatically.
- Eclipse Steady is a free, open source tool that analyses Java and Python projects for vulnerabilities and helps developers mitigate them.
- OWASP dependency-check: an open source vulnerability scanner.
- OSS Review Toolkit: an open source orchestrator able to collect security advisories for used dependencies from configured vulnerability data services.

Resources

- The MITRE's vulnerability database of CVEs. See also NIST's security database of NVDs, and satellite resources like CVE Details.
- Check also this new initiative from Google: the open source Vulnerabilities.
- The OWASP working group publishes a list of vulnerabilities scanners on their website, both from the commercial and open sources worlds.
- J. Williams and A. Dabirsiaghi. The unfortunate reality of insecure libraries, 2012.
- Detection, assessment and mitigation of vulnerabilities in open source dependencies, Serena Elisa Ponta, Henrik Plate & Antonino Sabetta, Empirical Software Engineering volume 25, pages 3175–3215(2020).
- A Manually-Curated Dataset of Fixes to Vulnerabilities of open source Software, Serena E. Ponta, Henrik Plate, Antonino Sabetta, Michele Bezzi, Cédric Dangremont. There is also a toolkit in development to implement the aforementioned dataset.

5.3 Manage software dependencies

Activity ID: GGI-A-23.

Description

A *dependency identification* program looks for the dependencies actually used within the codebase. As a result, the organisation must establish and maintain a list of known dependencies for its code base and watch the evolution of the identified providers.

Establishing and maintaining a list of known dependencies is an enabler for, and a prerequisite to:

- IP and licence checking: some licences cannot be mixed, even as a dependency. One has to know its dependencies to assess its associated legal risks.
- Vulnerabilities management: the entire piece of software is as weak as its smallest part: see the example of the Heartbleed flaw. One has to know its dependencies to assess its associated security risks.
- Lifecycle and sustainability: an active community on the dependency project is a bright sign for bug corrections, optimisations, and new features.
- Thoughtful selection of used dependencies, according to "maturity" criteria - the goal being to use open source components that are safe, with a sane and well-maintained codebase, and a living, active and reactive community that will accept external contributions, etc.

Opportunity Assessment

Identifying and tracking dependencies is a required step to mitigate the risks associated with any code reuse. In addition, implementing tools and processes to manage software dependencies is a prerequisite to properly manage quality, compliance, and security.

Consider the following questions:

- What is the company's risk (cost, reputation, etc.) if the software is corrupted, attacked or sued?
- Is the code base considered critical for people, the organisation, or business?
- What if a component upon which an application depends changes its repository?

The minimal and first step is to implement a software composition analysis (SCA) tool. Support by specialised consulting firms may be required for a full-fledged SCA or dependency mapping.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ Dependencies are identified in all in-house developed code.
- ☐ Dependencies are identified in all external code executed within the company.
- ☐ An easy-to-setup software composition analysis or dependency identification procedure is available for projects to add to their Continuous Integration process.
- ☐ Dependency analysis tools are used.

Tools

- OWASP Dependency check: dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within a project's dependencies.
- OSS Review Toolkit: a suite of tools to assist with reviewing Open Source Software dependencies.
- Fossa: fast, portable and reliable dependency analysis. Supports licence & vulnerability scanning. Language-agnostic; integrates with 20+ build systems.
- Software 360.
- Eclipse Dash license tool: takes a list of dependencies and requests ClearlyDefined to check their licences.
- The FOSSology Project: FOSSology is an open source project with the mission of advancing open source license compliance.

Recommendations

- Conduct regular audits about the dependencies and IP requirements to mitigate legal risks.
- Ideally, integrate dependencies management in the Continuous integration process so that issues (new dependency, licence incompatibility) are identified and fixed as soon as possible.
- Keep track of dependency-related vulnerabilities, keep users and developers informed.
- Inform people about the risks associated with wrong licencing.
- Propose an easy solution for projects to set up licence checking on their codebase.
- Communicate on its importance and help projects to add it to their CI systems.
- Set up a visible KPI for dependency-related risks.

Resources

- Existing OSS-licenced OSS licence compliance tools group page.
- Free and Open Source Software licence Compliance: Tools for Software Composition Analysis, by Philippe Ombredanne, nexB Inc.
- Software Sustainability Maturity Model.
- CHAOS: Community Health Analytics Open Source Software.

5.4 Manage key indicators

Activity ID: GGI-A-24.

Description

This activity collects and monitors a set of indicators that inform day-to-day managerial decisions and strategic options concerning professionally managed open source software.

Key metrics related to open source software form the backdrop of how well governance programmes are rolled out. The activity covers selecting a few indicators, publishing them to the teams and management, and sending regular updates on the initiative, e.g. via a newsletter or corporate news.

This activity requires:

- stakeholders to discuss and define the objectives of the program,
- the implementation of a measurement and data collection tool connected to the development infrastructure,
- the publication of at least one dashboard for the stakeholders and for all the people involved in the initiative.

Indicators are based on data that must be collected from relevant sources. Fortunately, there are plenty of sources for open source software engineering. Examples include:

- the development environment, the CI/CD production chain,
- the HR department,
- the testing and software composition analysis tools,

- the repositories.

Examples of indicators include:

- Number of resolved dependencies, displayed by licence type.
- Number of outdated/vulnerable dependencies.
- Number of licencing/ip issues detected.
- Contributions made to external projects.
- Bug open time.
- Number of contributors on a component, number of commits, etc.

This activity is about defining these requirements and measurement needs, and implementing a dashboard that shows in a simple and efficient manner the main indicators of the program.

Opportunity Assessment

Key indicators help understand and better manage the resources devoted to open source software, and measure the results in order to communicate effectively and reap the full benefits of the investment. By communicating broadly, more people can follow the initiative and will feel involved, ultimately making it an organisation-level concern and goal.

While each activity has evaluation criteria that help answer questions about progress achieved, there is still a need for monitoring done with numbers and quantitative indicators.

Whether in a small startup or a large global company, key metrics help keep teams focused and monitor performance. Metrics are crucial because they support decision-making and are the basis for monitoring decisions already taken.

With simple and practical numbers and graphics, the whole organisation members will be able to follow and synchronise efforts regarding open source, making it a shared concern and action. This also allows the various actors to better enter the course, contribute to the project and get the overall benefits.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ A list of metrics and how to collect them has been established.
- ☐ Tools to collect, store, process and display indicators are used.
- ☐ There is a generalised dashboard available to all participants that shows the progress made on the initiative.

Tools

- GrimoireLab from Bitergia.
- Generic BI tools (elasticsearch, grafana, R/Python visualisations…) are a good fit too, when the proper connectors are setup according to the defined goals.

Recommendations

- Write down the objectives and roadmap of the open source Governance.
- Communicate in-house about the actions and status of the initiative.
- Involve people in the definition of KPIs, in order to make sure that
  - they are well understood,
  - they provide a complete view of the needs and
  - they are considered and followed.
- Build at least one dashboard that can be displayed for everybody (e.g. on a screen in the room), with essential indicators to show the progress and overall situation.

Resources

- The CHAOSS community has many good references and resources related to open source indicators.

- Check out metrics for Project Attributes from the OW2 Market Readiness Levels methodology.
- A New Way of Measuring Openness: The Open Governance Index by Liz Laffan is an interesting reading about openness in open source projects.
- Governance Indicators: A Users' Guide is the UN's guide about governance indicators. Although it is applied to democracy, corruption and transparency of nations, the basics of measurement and indicators as applied to governance are well worth a read.

5.5 Run code reviews

Activity ID: GGI-A-44.

Description

Code review is a routine task involving manual and/or automated review of an application's source code before releasing a product or delivering a project to the customer. In the case of open-source software, code review is more than just about catching errors opportunistically; it is an integrated approach to collaborative development carried out at the team level.

Code reviews should apply to code developed in-house as well as to code reused from external sources, as it improves general confidence in code and reinforces ownership. It is also an excellent way to enhance global skills and knowledge within the team and foster team collaboration.

Opportunity Assessment

Code reviews are valuable whenever the organisation develops software or reuses external pieces of software. While being a standard step in the software engineering process, code reviews in the context of open-source bring specific benefits such as:

- When publishing internal source code, verify adequate quality guidelines are respected.
- When contributing to an existing open source project, verify that guidelines of the targeted project are respected.
- The publicly available documentation is updated accordingly.

It is also an excellent opportunity to share and enforce some of your company legal compliance policy rules, such as:

- Never remove existing licence headers or copyrights found in reused open-source code.
- Do not copy & paste source code from Stack Overflow without prior permission from the legal team.
- Include the correct copyright line when required.

Code reviews will bring trust and confidence to code. If people are not sure about the quality or potential risks of using a software product, they should conduct peer- and code- reviews.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ Open source code review is recognised as a necessary step.
- ☐ Open source code reviews are planned (either regularly or at critical moments).
- ☐ A process for conducting open-source code reviews has been collectively defined and accepted.
- ☐ Open-source code reviews are a standard part of the development process.

Recommendations

- Code review is a collective task that works better in a good collaborative environment.
- Do not hesitate to use existing tools and patterns from the open-source world, where code reviews have been a standard for years (decades).

Resources

- What is Code Review?: a didactic read on code review found on Red Hat's Open Practice Library.
- Best Practices for Code Reviews: another interesting perspective on what code review is about.

6 Culture goal activities

6.1 Promote open source development best practices

Activity ID: GGI-A-25.

Description

This activity is about defining, actively promoting and implementing open source best practices within the development teams.

As a starting point the following topics might be considered for attention:

- User and developer documentation.
- Proper organisation of the project on a publicly accessible repository.
- Promote and implement controlled reuse.
- Providing a complete and up-to-date product documentation.
- Configuration Management: git workflows, collaborative patterns.
- Release management: release early & release often, stable vs development versions, etc.

OSS projects have a special, bazaar-like modus operandi. In order to allow and foster this collaboration and mindset, some practices are recommended that facilitate collaborative and decentralised development and contributions from third-party developers...

Community documents Make sure that all projects within the company propose the following documents:

- README -- quick description of the project, how to interact, links to resources.
- Contributing -- introduction for people willing to contribute.
- Code Of Conduct -- What is acceptable -- or not -- as behaviour within the community.
- LICENSE -- the default licence of the repository.

REUSE best practices REUSE is an initiative from the Free Software Foundation Europe to improve reuse of software and streamline OSS and licence compliance.

Opportunity Assessment

Although it heavily depends on the OSS common-knowledge among the team, training people and creating processes that enforce these practices is always beneficial. It is even more important when:

- potential users and contributors are not known,
- developers are not used to open source development.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ Project sets a list of open source best practices to comply with.
- ☐ Project monitors its alignment with best practices.
- ☐ Development team has built awareness about complying with OSS best practices.
- ☐ New best practices are regularly evaluated, and an effort is made to implement them.

Tools

- The REUSE helper tool assists with making a repository conformant with the REUSE best practices. It can be included in many development processes to confirm the current status.
- ScanCode has the ability to list all community and legal documents in the repository: see feature description.
- GitHub has a nice feature to check for missing community documents. It can be found in the Repository page > "Insights" > "Community".

Recommendations

- The list of best practices depends on the context and domain of the program and should be re-evaluated regularly in a continuous improvement manner. Practices should be monitored and regularly assessed to track down progress.
- Train people about OSS reuse (as consumers) and ecosystems (as contributors).
- Implement REUSE.software as in activity #14.
- Set up a process to manage legal risks associated with reuse and contributions.
- Explicitly encourage people to contribute to external projects.
- Provide a template or official guidelines for project structure.
- Set up automated checks to make sure that all projects comply with the guidelines.

Resources

- OW2's list of open source best practices from the Market Readiness Levels assessment methodology.
- REUSE's official website with specification, tutorial, and FAQ.
- GitHub's community guidelines.
- An example of configuration management best practices using GitHub.

6.2 Contribute to open source projects

Activity ID: GGI-A-26.

Description

Contributing to open source projects that are freely used is one of the key principles of good governance. The point is to avoid being a simple passive consumer and give back to the projects. When people add a feature or fix a bug for their own purpose, they should make it generic enough to contribute to the project. Developers must be allowed time for contributions.

This activity covers the following scope:

- Working with upstream open source projects.
- Reporting bugs and feature requests.
- Contributing code and bug fixes.
- Participating in community mailing lists.
- Sharing experience.

Opportunity Assessment

The main benefits of this activity are:

- It increases the general knowledge and commitment to open source within the company, as people start contributing and get involved in open source projects. They get a feeling of public utility and improve their personal reputation.
- The company increases its visibility and reputation as contributions make their way through the contributed project. This shows that the company is actually involved in open source, contributes back, and promotes fairness and transparency.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

☐ There is a clear and official path for people willing to contribute.
☐ Developers are encouraged to contribute back to open source projects they use.
☐ A process is in place to ensure legal compliance and security of contributions by developers.

- KPI: Volume of external contributions (code, mailing lists, issues..) by individual, team, or entity.

Tools

It may be useful to follow contributions, both to keep track of what is contributed and to be able to communicate on the company's effort. Dashboards and activity tracking software can be used for this purpose. Check:

- Bitergia's GrimoireLab
- ScanCode

Recommendations

Encourage people within the entity to contribute to external projects, by:

- Allowing them time to write generic, well-tested bug fixes and features, and to contribute them back to the community.
- Providing training to people about contributing back to open source communities. This is both about technical skills (improving your team's knowledge) and community (belonging to the open source communities, code of conduct, etc.).
- Provide training on legal, IP, technical issues, and set up a contact within the company to help with these topics if people have doubts.
- Provide incentives for published work.
- Note that contributions from the company/entity will reflect its code quality and involvement, so make sure your development team provides code that is good enough.

Resources

- The CHAOSS initiative from the Linux Foundation has some tools and pointers about how to track contributions in development.

6.3 Belong to the open source community

Activity ID: GGI-A-27.

Description

This activity is about developing among developers a feeling of belonging to the greater open source community. As with every community, people and entities have to participate and contribute back to the whole. It reinforces the links between practitioners and brings sustainability and activity to the ecosystem. On a more technical side, it allows choosing the priorities and roadmap of projects, improves the level of general knowledge and technical awareness.

This activity covers the following:

- **Identify events** worth attending. Connecting people, learning about new technologies and building a network are key factors to get the full benefits of open-source.
- Consider **foundation memberships**. Open-source foundations and organisations are a key component of the open-source ecosystem. They provide technical and organisational resources to projects, and are a good neutral place for sponsors to discuss common issues and solutions, or work on standards.
- Watch **working groups**. Working groups are neutral collaborative workspaces where experts interact on a specific domain like IoT, modelling or science. They are a very efficient and cost-effective mechanism to tackle common, albeit domain-specific concerns together.
- **Budget participation**. At the end of the journey, money is the enabler. Plan required expenses, allow people paid time for these activities, anticipate next moves, so the program doesn't have to stop after a few months short of funding.

Opportunity Assessment

Open source works best when done in relation with the open source community at large. It facilitates bug fixing, solution sharing, etc.

It is also a good way for companies to show their support to open-source values. Communicating about the company's involvement is important both for the company's reputation and for the open-source ecosystem.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ A list of events people could attend is drafted.
- ☐ There is a monitoring of public talks given by team members.
- ☐ People can submit event participation requests.
- ☐ People can submit projects for sponsorship.

Recommendations

- Survey people to get to know what events they like or would be the most beneficial for their work.
- Set up in-house communications (newsletter, resource centre, invitations…) so people know about the initiatives and can participate.
- Make sure that these initiatives can benefit various types of people (developers, administrators, support…), not only C-level executives.

Resources

- What motivates a developer to contribute to open source software? An article by Michael Sweeney on clearcode.cc.
- Why companies contribute to open source An article by Velichka Atanasova from VMWare.
- Why your employees should be contributing to open source A good read by Robert Kowalski from CloudBees.
- 7 ways your company can support open source An article from Simon Phipps for InfoWorld.
- Events: the life force of open source An article by Donna Benjamin from RedHat.

6.4 HR perspective

Activity ID: GGI-A-28.

Description

Switching to open source culture has deep HR impacts:

- **New processes and contracts**: Contracts have to be adapted to allow and promote external contributions. This includes IP and licensing issues for work done within the company, but also the ability for the employee or contractor to have their own projects.
- **Different types of people**: People working with open source often have different incentives and mindsets than with pure proprietary, corporate people. Processes and mindsets need to adapt to this community reputation-oriented paradigm, in order to attract new types of talent and retain them along.
- **Career development**: need to offer a career path that nurtures and values employees for their technical and soft skills as well as the competencies expected by your organisation (collaboration to drive community efforts, communication to act as spokesperson for your company, etc.). By all means, HR has a key role in enabling open source as a cultural goal.

Workforce For a developer who has been working on the same proprietary solution for a long time, switching to open source may seem quite a change, and require adaptation. But for most developers, open source software only brings benefits.

Developers getting out of school or university today have all always been working on open source. Within a company, the large majority of developers are using open source languages and importing open source libraries or snippets every day. It is indeed much easier to paste lines of open source code in a program than to trigger the internal sourcing process, which escalates through multiple validations through the management line.

Open source makes the job of the developer more interesting because with open source, a developer is always on the lookout to find out what their peers outside the company have been inventing, and therefore remains on the cutting edge of the technology.

For an organisation, there needs to be an HR strategy to 1/ skill or re-skill the existing workforce 2/ reflect and position the company on hiring new talents hence what is the attractiveness of the company when it comes to open source.

> Getting people with a good FLOSS mindset, who already understand the code, and know how to work well with others is wonderful. The alternative of evangelising / training / interning is worth doing but more expensive & time consuming.
>
> — OSS Software Vendor CEO

This illustrates that hiring people with open source DNA is an acceleration path to consider in HR strategy.

Processes

- Establish or revisit job descriptions (technical skills, soft skills, competencies and experiences)
- Training programs: self-training, formal training, management coaching, peer mapping, communities
- Establish or revisit career path: competencies, key results/impact and career steps

Opportunity Assessment

1. Frame development practices: the problem is probably not so much to spur developers to use more open source, but rather to make sure that they use it safely, in compliance with the licensing terms of each open source technology, and without abandoning traditional security checks (open source lines of code could contain malicious codes),
2. Revisit collaboration practices: with development practices, the opportunity is to extend the agility and collaboration to other lines of business in your organisation. Inner sourcing is often used to foster these behaviours, though this might be half of the way to open source culture,
3. Organisation's culture: in the end, this is all about your organisation's culture: open source can be the flagship for values such as openness, collaboration, ethics, sustainability.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ Training is available for presenting both the benefits and the constraints (Intellectual Property licensing terms compliance) related to open source.
- ☐ Every developer, every architect, every project leader (or Product Owner/Business Owner), understands the benefits and the constraints (Intellectual Property licensing terms compliance) related to open source.
- ☐ Developers are encouraged to contribute to open source communities, and take responsibility for them, and could receive adequate training to do it.
- ☐ Skills and competencies are reflected in organisation job descriptions and career steps.
- ☐ The experience that developers acquired in open source (contributions to open source communities, participation in the internal compliance process, external spokespersons for the company, ...) is taken into account in the HR evaluation process.

Tools

- Skills matrix.
- Public training programs (ex. open source school).
- Sourcing: GitHub, GitLab, LinkedIn, Meetups, Epitech, Epita...
- Contract templates (Loyalty clause).
- Job descriptions (templates) & career steps (templates).

Recommendations

Most of the time nowadays, developers already know some open source principles and are willing to work with, and on, open source software. However, there are still a few actions that the management should take:

- Preference for OSS experience in hiring, even though the job for which the developer is being hired only relates to proprietary technology. Chances are, with the digital transformation, that the developer will someday have to work on open source.
- OSS training program: Every developer, every architect, every project leader (or Product Owner/Business Owner), should have access to training resources (videos or face-to-face training) that present the benefits of open source and also the constraints in terms of Intellectual Property and licensing compliance.
- Training should be made available for developers who want to contribute to open source communities and be part of the governance bodies of these communities (Linux certifications).
- Recognition in the HR personal assessment processes of the contribution of the employee (developer or architect) to open source related topics such as contributions to open source communities and compliance with Intellectual Property licensing terms. Most topics are shared and fit into technical career paths, whereas some might or should be specific.
- Best kept secret and company posture: need to address the communication aspects (how core this is to your organisation that it might be reflected in your annual report), how does it impact your communication posture (an open source contributor could be a spokesperson for your company including press contacts).

Resources

- Regarding the ability of people to speak outside the company during events, please see Activity 31: "(Engagement Goal) Publicly asserting the use of open source".

6.5 Upstream first

Activity ID: GGI-A-39.

Description

This activity is concerned with developing awareness with regard to the benefits of contributing back and enforcing the upstream first principle.

With the upstream first approach all development on an open source project is to be made with the level of quality and openness required to be submitted to a project's core developers and published by them.

Opportunity Assessment

Writing code with upstream first in mind results in:

- better quality code,
- code that is ready to be submitted upstream,
- code that is merged in the core software,
- code that will be compatible with future version,
- recognition by the project community and better and more profitable cooperation.

> Upstream First is more than just "being kind". It means you have a say in the project. It means predictability. It means you are in control. It means you act rather than react. It means you understand open source. (Maximilian Michels)

Progress Assessment

The following **verification points** demonstrate progress in this Activity: Upstream first is implemented?

- ☐ Significant increase in the number of pull/merge requests submitted to third party projects.

☐ A list of third party projects for which upstream first must be applied has been drafted.

Recommendations

- Identify developers with most experience at interacting with upstream developers.
- Facilitate interaction between developers and core developers (events, hackathons, etc.)

Resources

- A clear explanation of the Upstream First principle and why it fits in the Culture Goal: https://maximilianmichels.com/2021/upstream-first/.

  Upstream First means that whenever you solve a problem in your copy of the upstream code which others could benefit from, you contribute these changes back upstream, i.e. you send a patch or open a pull request to the upstream repository.
- What is Upstream and Downstream in Software Development? A crystal clear explanation.
- Explained from the Chromium OS design documents: Upstream First.
- Red Hat on upstream and the advantages of upstream first.

7 Engagement goal activities

7.1 Engage with open source projects

Activity ID: GGI-A-29.

Description

This activity is about committing significant contributions to some OSS projects that are important to you. Contributions are scaled up and committed at the organisation level (not personal level as in #26). They can take several forms, from direct funding to resources allocation (e.g. people, servers, infrastructure, communication, etc.), as long as they benefit the project or ecosystem sustainably and efficiently.

This activity is a follow-up on activity #26 and brings open source projects' contributions to the level of the organisation, making them more visible, powerful, and beneficial. In this activity, the contributions are supposed to bring a substantial, long-term improvement to the OSS project: e.g., a developer or team who develops a much-wanted new feature, infrastructure assets, servers for a new service, take-over of the maintenance of a widely used branch.

The idea is to set aside a percentage of resources to sponsor open source developers that write and maintain libraries or projects that we use.

This activity implies to have a mapping of open source software used, and an evaluation of their criticality to decide which one to support.

Opportunity Assessment

> If every company using open source contributed at least a little, we would have a healthy ecosystem. https://news.ycombinator.com/item?id=25432248

Supporting projects helps ensure their sustainability and provides access to information, even maybe help influence and prioritise some developments (although this should not be the main reason for supporting projects).

Potential benefits of this activity: ensuring bug reports are prioritised and developments are integrated into the stable version. Possible costs associated with the activity: committing time to projects, cash commitment.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ Beneficiary project identified.
- ☐ Support option decided, such as direct monetary contribution or code contribution.
- ☐ Task leader appointed.
- ☐ Some contribution has happened.
- ☐ Result of contribution has been evaluated.

Verification points borrowed from OpenChain self certification questionnaire:

- ☐ We have a policy for contribution to open source projects on behalf of the organisation.
- ☐ We have a documented procedure governing open source contributions.
- ☐ We have a documented procedure for making all Software Staff aware of the open source contribution policy.

Tools

Some organisations offer mechanisms for funding open source projects (it could be convenient if your target project is in their portfolios).

- Open Collective.
- Software Freedom Conservancy.
- Tidelift.

Recommendations

- Concentrate on projects that are critical for the organisation: these are the projects you most want to help with your contributions.
- Target community projects.
- This activity requires a minimum familiarity with a target project.

Resources

- How to support open source projects now: A short page with ideas on funding open source projects.
- Sustain OSS: a space for conversations about sustaining open source

7.2 Support open source communities

Activity ID: GGI-A-30.

Description

This activity is about engaging with institutional representatives of the open source world.

It is achieved through:

- Joining OSS foundations (including the financial cost of membership).
- Supporting, advocating foundations activities.

This activity involves allocating the development and IT teams some time and budget to participate in open source communities.

Opportunity Assessment

Open source communities are at the forefront of the evolution of the open source ecosystem. Engaging with open source communities has several advantages:

- it helps keep informed and up to date,
- it enhances the profile of the organisation,
- membership comes with benefits,
- it provides additional structure and motivation to the open source IT team.

Costs include:

- membership fees,
- personnel time and some travel budget allocated to participate in community activities,
- monitoring of IP commitment.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ The organisation is a signed member of an open source foundation.
- ☐ The organisation participates in the governance.
- ☐ Software developed by the organisation is submitted to / has been added to the code base of a foundation.
- ☐ Membership is acknowledged on the websites of both the organisation and the community.
- ☐ Performed cost/benefit assessment of the membership.
- ☐ A contact point for the community has been appointed.

Recommendations

- Join a community compatible with your size and resources, i.e. a community that can hear your voice and where you can be a recognized contributor.

Resources

- Check out this useful page from the Linux Foundation on the why and how to join an open source community.

7.3 Publicly assert use of open source

Activity ID: GGI-A-31.

Description

This Activity is about acknowledging the use of OSS in an information system, in applications and in new products.

- Providing success stories.
- Presenting at events.
- Funding participation to events.

Opportunity Assessment

It is now generally accepted that most information systems run on OSS, and that new applications are for the most part made by reusing OSS.

The main benefit of this activity is to create a level playing field between OSS and proprietary software, to make sure OSS is paid equal attention to and managed just as professionally as proprietary software.

A side benefit is that it greatly helps raise the profile of the OSS ecosystem and, since OSS users are identified as ”innovators” it also enhances the attractiveness of the organisation.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ Commercial open source vendors are granted authorization to use the organisation’s name as customer reference.
- ☐ Contributors are allowed to do so and express themselves under the organisation’s name.
- ☐ Use of OSS is openly mentioned in the IT department annual report.
- ☐ There is no obstacle to the organisation explaining their use of OSS in the media (interviews, OSS and industry events, etc.).

Recommendations

- The objective of this activity is not for the organisation to become an OSS activism body, but to make sure there is no obstacle to the public recognising its use of OSS.

Resources

- Example of CERN publicly asserting their use of OpenStack

7.4 Engage with open source vendors

Activity ID: GGI-A-33.

Description

Secure contracts with open source vendors that provide software critical to you. Companies and entities that produce open source software need to thrive to provide maintenance and development of new features. Their specific expertise is required on the project, and the community of users relies on their continued business and contributions.

Engaging with open source vendors takes several forms:

- Subscribing support plans.
- Contracting local service companies.
- Sponsoring developments.
- Paying for a commercial licence.

This activity implies considering open source projects as fully-featured products worth paying for, much like any proprietary products -- although usually far less expensive.

Opportunity Assessment

The objective of this activity is to ensure professional support of open source software used in the organisation. It has several benefits:

- Continuity of service through timely bug fixes.
- Service performance through optimised installation.
- Clarification of the legal/commercial status of the software used.
- Access to early information.
- Stable budget forecast.

The cost is obviously that of the support plans selected. Another cost might be to depart from bulk outsourcing to large systems integrators in favour of fine-grained contracting with expert SMEs.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

☐ Open source used in the organisation is backed by commercial support.
☐ Support plans for some open source projects have been contracted.
☐ Cost of open source support plans is a legitimate entry in the IT budget.

Recommendations

- Whenever possible, find local expert SMEs.
- Beware of large systems integrators reselling third-party expertise (reselling support plans that expert open source SMEs actually provide).

Resources

A couple of links illustrating the commercial reality of open source software:

- An investor's view of the community to business evolution of open source projects.
- A quick read to understand commercial open source.

7.5 Open source procurement policy

Activity ID: GGI-A-43.

Description

This activity is about implementing a process to select, acquire, purchase open source software and services. It is also about considering the actual cost of open source software and provisioning for it. OSS may be "free" at first sight, but it is not without internal and external costs such as integration, training, maintenance and support.

Such policy requires that both open source and proprietary solutions are symmetrically considered when evaluating value for money as the optimum combination of the total cost of ownership and quality. Therefore, the IT Procurement department should actively and fairly consider open source options, while at the same time ensuring proprietary solutions are considered on an equal footing in purchasing decisions.

Open source preference can be explicitly stated based on the intrinsic flexibility of the open source option when there is no significant overall cost difference between proprietary and open source solutions.

Procurement departments must understand that companies offering support for OSS typically lack the commercial resources to participate in procurement competitions, and adapt their open source procurement policies and processes accordingly.

Opportunity Assessment

Several reasons justify the efforts to set up specific open source procurement policies:

- Supply of commercial open source software and services is growing and cannot be ignored, and requires the implementation of dedicated procurement policies and processes.
- There is a growing supply of highly competitive commercial open source business solutions for corporate information systems.
- Even after adopting a free-of-charge OSS component and integrating it into an application, internal or external resources must be provided to maintain that source code.
- Total Cost of Ownership (TCO) is often (although not necessarily) lower for FOSS solutions: no licence fees to pay when purchasing/upgrading, open market for service providers, option to provide some or all of the solution yourself.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ New call for proposals proactively request open source submissions.
- ☐ Procurement department has a way to evaluate open source vs proprietary solutions.
- ☐ A simplified procurement process for open source software and services has been implemented and documented.
- ☐ An approval process drawing from cross-functional expertise has been defined and documented.

Recommendations

- ”Be sure to tap into the expertise of your IT, DevOps, cybersecurity, risk management, and procurement teams when creating the process.” (from 5 Open Source Procurement Best Practices).
- Competition law may require that ”open source” not be specifically mentioned.
- Select technology upfront then go to RFP for customisation and support services.

Resources

- Decision factors for open source software procurement: not new, but still a great read by our colleagues at OSS-watch in the UK. Check out the slides.
- 5 Open Source Procurement Best Practices: a recent piece on open source procurement with useful hints.

8 Strategy goal activities

8.1 Setup a strategy for corporate open source governance

Activity ID: GGI-A-16.

Description

Defining a high-level strategy for open source governance within the company ensures the consistency and visibility of approaches towards both in-house usage and external contributions and involvement. It makes the company's communication more effective by offering a clear and established vision and leadership.

The shift towards open source brings with it numerous benefits, as well as some duties and a change in the company's culture. It can impact business models and influence the manner in which an organisation presents its value and offer, and in its position towards its customers and competitors.

This activity includes the following tasks:

- Set up an OSS Officer, with (top) management sponsoring and backing.
- Set up and publish a clear roadmap for open source, with stated objectives and expected benefits.
- Make sure that all top-level management knows about it and acts in accordance with it.
- Promote OSS inside the company: encourage people to use it, foster in-house initiatives and level of knowledge.
- Promote OSS outside the company: through official statements and communication, and visible involvement in OSS initiatives.

Defining, publishing and enforcing a clear and consistent strategy also helps buy-in from all people within the company and eases further initiatives from teams.

Opportunity Assessment

It's a good time to work on this activity if:

- There is no coordinated effort from management, and open source is still seen as an ad-hoc solution.
- There are already in-house initiatives, but they don't penetrate up to the upper levels of management.
- The initiative was started some time ago but faces many obstacles, and still doesn't yield the expected results.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is a clear open source governance charter for the company. The charter should contain:
  - what to achieve,
  - who we do this for,
  - what the power of the strategist(s) is and what not.
- ☐ An open source roadmap is widely available and accepted throughout the company.

Recommendations

- Set up a group of people and processes to define and monitor open source governance within the company. Ensure there is a clear commitment from the top-level management to the open source initiatives.
- Communicate about the open source strategy within the organisation, make it a major concern and a true corporate commitment.
- Ensure that the roadmap and strategy is well understood by everybody, from development teams to management and infrastructure staff.

- Communicate on its progress, so people know where the organisation is regarding its commitment. Publish regular updates and indicators.

Resources

- Checklist and references for Open Governance.
- L'open source comme enjeu de souveraineté numérique, by Cédric Thomas, OW2 CEO, Workshop at Orange Labs, Paris, January 28, 2020 (french only).
- A series of guides to manage open source within the enterprise, by the Linux Foundation.
- A fine example of open source strategy document, by the LF Energy group

8.2 C-Level awareness

Activity ID: GGI-A-34.

Description

The open source initiative of the organisation will yield its strategic benefits only if it is enforced at its highest levels by integrating the open source DNA into the company's strategy and internal working. Such commitment cannot happen if higher-level executives and top management are not themselves a part of it. The training and open source mindset must also be extended to those who shape the policies, decisions and overall strategy, both inside and outside the company.

This commitment ensures that practical improvements, mindset changes and new initiatives are met with consistent, benevolent and sustainable support from the hierarchy, bringing in more fervent participation from workers. It shapes how external actors see the organisation, bringing in reputational and ecosystem benefits. It is also a means to establish the initiative and its benefits in the mid and long term.

Opportunity Assessment

This activity becomes essential if/when:

- The organisation has set global goals relevant to open source management, but struggles to achieve them. It is unlikely that the initiative can achieve anything without good knowledge and a clear commitment from higher-level executives.
- The initiative has already started and is making progress, but the higher levels of the hierarchy do not follow it up properly.

Hopefully, it should become evident that anything but ad hoc usage of open source requires a consistent and well-thought approach, given the range of teams and cultural change it can bring.

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is a mandated governance office/officer empowered to set a uniform open source strategy across the company and ensure that the scope is clear.
- ☐ There is a clear, binding commitment from the hierarchy to the OSS strategy.
- ☐ There is transparent communication by the hierarchy about its commitment to the program.
- ☐ The hierarchy is available to discuss open source software. It can be solicited and challenged on its promises.
- ☐ There is an appropriate budget and funding for the initiative.

Recommendations

Examples of actions associated with this activity include:

- Conduct training to demystify OSS to C-level management.
- Obtain explicit, practical endorsement for OSS usage and strategy.
- Explicitly mention and endorse the OSS program in internal communications.
- Explicitly mention and endorse the OSS program in public communications.

Open source is a *strategic enabler* that embarks *enterprise culture*. What does this mean?

- Open source can be leveraged as a mechanism to disrupt suppliers and reduce software acquisition costs.
  - Should open source come under the purview of *Software Asset Managers* or *purchasing departments*?
- Open source licences enshrine the freedoms that deliver the benefits of open source, but they also carry *obligations*. If not met appropriately, responsibilities can create legal, commercial and image risks to an organisation.
  - Will licence conditions grant visibility into areas of code that should remain confidential?
  - Will it impact my organisation's patent portfolio?
  - How should project teams be trained and supported on this subject?
- Contributing back to external open source projects is where the biggest value of open source lies.
  - How should my company encourage (and track) this?
  - How should developers use GitHub, GitLab, Slack, Discord, Telegram, or any of the other tools open source projects habitually use?
  - Can open source impact the company's HR policies?
- Of course, it's not all about contributing back, what about my own open source projects?
  - Am I ready to do *open* innovation?
  - How will my projects manage *incoming* contributions?
  - Should I spend the effort to nurture a community for a given project?
  - How should I lead the community, what role should community members have?
  - Am I ready to cede roadmap decisions to a community?
  - Can open source be a valuable tool to reduce silo-isation between company teams?
  - Do I need to handle open source transfer from one company entity to another?

8.3 Open source and digital sovereignty

Activity ID: GGI-A-35.

Description

Digital sovereignty can be defined as the

> "Ability and opportunity of individuals and institutions to execute their role(s) in the digital world independently, intentionally and safely." — Competence Centre for Public IT, Germany

In order to properly conduct its business, any entity has to rely on some other partners, services, products and tools. Reviewing the ties and constraints of these dependencies enable the organisation to assess and control its dependence towards external factors, thus improving its autonomy and resilience.

As an example, vendor lock-in is a strong factor of dependence that may impede the organisation's processes and added value and as such, it should be avoided. Open source is one of the ways out of this lock. Open source plays a significant role in digital sovereignty, allowing a greater choice between solutions, providers and integrators, and greater control over IT roadmaps.

It should be noted that digital sovereignty is not a trust issue: we obviously need to trust our partners and providers, but the relationship gets even better when it's based on mutual consent and recognition, rather than forced contracts and strains.

Here are some advantages of a better digital sovereignty:

- Improve the ability of the organisation to make its own choices without constraints.
- Improve the resilience of the company regarding external actors and factors.
- Improve negotiating position when dealing with partners and service providers.

Opportunity Assessment

- How difficult/expensive is it to move away from a solution?

- Could the solution providers impose unwanted conditions on their service (e.g. licence change, contracts updates)?
- Could the solution providers unilaterally increase their prices, simply because we do not have a choice?

Progress Assessment

The following **verification points** demonstrate progress in this Activity:

- ☐ There is an assessment of critical dependencies for the organisation's providers and partners.
- ☐ There is a backup plan for these identified dependencies.
- ☐ There is a stated requirement for digital sovereignty when new solutions are investigated.

Recommendations

- Identify key dependency risks from service providers and 3rd party entities.
- Maintain a list of open-source alternatives to critical services.
- Add a requirement when selecting new tools and services used within the entity, stating the need for digital sovereignty.

Resources

- A Primer on Digital Sovereignty & Open Source: part I and A Primer on Digital Sovereignty & Open Source: part II, from the Open-Sourcerers website.
- An excellent superuser.openstack.org article on The Role of Open Source in Digital Sovereignty. Here is a short extract:

  > Digital Sovereignty is a key concern for the 21st century, especially for Europe. Open source has a major role to play in enabling digital sovereignty, by allowing everyone to access the necessary technology, but also by providing the governance transparency and interoperability necessary for those solutions to succeed.

- The European Union's take on digital sovereignty, from the Open Source Observatory (OSOR): Open Source, digital sovereignty and interoperability: The Berlin Declaration.
- The UNICEF's position on Open Source for Digital Sovereignty.

8.4 Open source enabling innovation

Activity ID: GGI-A-36.

Description

> Innovation is the practical implementation of ideas that result in the introduction of new goods or services or improvement in offering goods or services.
>
> — Schumpeter, Joseph A.

Open source can be a key factor for innovation through diversity, collaboration and a fluent exchange of ideas. People from different backgrounds and domains may have different perspectives and provide new, improved or even disruptive answers to known problems. One can enable innovation by listening to different views and actively promoting open collaboration on projects and topics.

Similarly, participating in the elaboration and implementation of open standards is a great promoter of good practices and ideas to improve the company's daily work. It also allows the entity to drive and influence innovation to where and what it needs, and enhances its global visibility and reputation.

Through innovation, open source makes it possible not only to transform the goods or services that your company markets, but also to create or modify the whole ecosystem in which your company wants to thrive.

As an example, by releasing Android as open source, Google is inviting hundreds of thousands of companies to build up their own services based upon this open source technology. Google is thus creating a whole ecosystem from which all participants could benefit. Of course, very few companies are powerful enough to create an ecosystem by their own decision. But there are many examples of alliances between companies to create such an ecosystem.

Opportunity Assessment

It is important to assess the position of your company compared with its competitors and its partners and customers because it would often be risky for a company to drift too far away from the standards and technologies used by its customers, partners and competitors. Innovation obviously means being different, but what differs should not represent too large a scope; otherwise, your company would not benefit from the software developments made by the other companies of the ecosystem and from the business momentum the ecosystem provides.

Progress Assessment

The following **verification points** demonstrate progress in this activity:

- ☐ The technologies -- and open source communities that develop them -- that have an impact on the business have been identified.
- ☐ The progress and publications of these open source communities are monitored -- I am even aware of their strategy before the releases are made public.
- ☐ Employees of the organisation are members of (some of) these open source communities and influence their roadmaps and technical choices by contributing lines of codes and participating in the governance bodies of these communities.

Recommendations

Out of all the technologies that are necessary to run your business, you should identify:

- the technologies that could be the same as your competitors,
- the technologies that should be specific to your company.

Stay up-to-date on emerging technologies. Open source has been driving innovation for the last decade, and many day-to-day powerful tools come from there (think of Docker, Kubernetes, Apache Big Data projects, or Linux). No need to know everything about everything, but one should know enough of the state of the art to identify interesting new trends.

Allow, and encourage, people to submit innovative ideas, and to bring them forward. If possible, spend resources on these initiatives and make them grow. Rely on people's passion and will to create and foster emerging ideas and trends.

Resources

- 4 innovations we owe to open source.
- The Innovations of Open Source, from Professor 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.

8.5 Open source enabling digital transformation

Activity ID: GGI-A-37.

Description

> "Digital Transformation is the adoption of digital technology to transform services or businesses, through replacing non-digital or manual processes with digital processes or replacing older digital technology with newer digital technology." (Wikipedia)

When the most advanced organisations in Digital Transformation jointly drive change through their Business, IT and Finance to anchor digital in the way, they reconsider:

- Business model: value chain with ecosystems, as a service, SaaS.
- Finance: opex/capex, people, outsourcing.
- IT: innovation, legacy/asset modernization.

Open source is at the heart of digital transformation:

- Technologies, Agile practices, product management.
- People: collaboration, open communication, development/decision cycle.
- Business models: try & buy, open innovation.

In terms of competitiveness, the most visible processes are probably the processes that directly impact the customer experience. And we have to recognize that the big players, as well as start-up companies, by delivering totally unprecedented customer experience, drastically changed customer expectations.

Customer experience as well as all the other processes within a company entirely depend on IT. Every company has to transform its IT, this is what the digital transformation is about. Companies that have not done it yet, have now to achieve their digital transformation as fast as possible, otherwise the risk is that they could be wiped out of the market. Digital Transformation is a condition for survival. Since the stakes are so high, a company cannot entirely leave the digital transformation to a supplier. Every company has to get hands on with its IT, which means that every company has to get hands on with open source software because there is no IT without open source software.

Expected benefits of the digital transformation include:

- Simplify, automate core processes, make them real-time.
- Enable fast responses to competitive changes.
- Take advantage of Artificial intelligence and big data.

Opportunity Assessment

Digital transformation could be managed by:

- Segments of the IT: Production IT, Business Support IT (CRM, billing, procurement…), Support IT (HR, Finance, accounting…), Big Data.
- Type of technology or process supporting the IT: Infrastructure (cloud), Artificial Intelligence, Processes (Make-or-Buy, DevSecOps, SaaS).

Injecting open source in a particular segment or technology of your IT reveals that you want to get hands on in this segment or technology, because you assessed that this particular segment or technology of your IT is important for the competitiveness of your company. It is important to assess the position of your company compared not only with your competitors, but also with other industries, and key players in terms of customer experience and market solutions.

Progress Assessment

1. Level 1: Situation assessment

- I have identified:
  - the segments of IT that are important for the competitiveness of my company, and
  - the open source technologies required to develop applications in these segments.
- And I have thus decided:
  - on which segments I want to manage in-house the development of projects, and
  - on which open source technologies I need to build in-house expertise.

2. Level 2: Engagement

- On some selected open source technologies used within the company, several developers have been trained and are recognized as valuable contributors by the open source community.

In some selected segments, projects based upon open source technologies have been launched.

3. Level 3: Generalisation

- For all projects, an open source alternative is systematically being investigated during the inception stage of the project. To make it easier for the project team to study such open source alternative, a central budget, and a central team of architects, hosted in the IT Department, is dedicated to providing assistance to the projects.

KPIs:

- KPI 1. Ratio for which an open source alternative was investigated: (Number of projects / Total number of projects).
- KPI 2. Ratio for which the open source alternative was chosen: (Number of projects / Total number of projects).

Recommendations

Beyond the headline, Digital Transformation is a mindset that involves some fundamental changes, and this should also (or even mainly) come from the top-level layers of the organisation. Management shall promote initiatives, new ideas, manage risks, and potentially update existing procedures to make them fit new concepts.

Passion is a huge factor of success. One of the means developed by key players in the field is to set up open spaces for new ideas, where people can submit, and freely work on, their ideas about digital transformation. Management should encourage such initiatives.

Resources

- Eclipse Foundation: Enabling Digital Transformation in Europe Through Global Open Source Collaboration.
- Europe: Open source software strategy.
- Europe: Open source software strategy 2020-2023.

9 Conclusion

As we have said before, open source good governance is not a destination; it's a journey. We need to care about our common assets, about the communities and ecosystem that make it strive, because our own common, and thus individual, success depends on it.

We, as software practitioners and open source enthusiasts, are committed to continue improving the Good Governance Initiative handbook and work on its dissemination and reach. We strongly believe that organisations, individuals, and communities need to work hand in hand to build a better and greater set of commons, available and beneficial to all.

You are welcome to join the OSPO Alliance, contribute to our work, spread the word, and be the ambassador of a better open source awareness and governance within your own ecosystem. There are a breadth of resources available out there, from blog posts and research articles to conferences and online training courses. We also provide a set of useful material on our website, and we are happily willing to help as much as we can.

Let's define and build together the future of the Good Governance Initiative!

9.1 Contact

The preferred way to get in touch with the OSPO Alliance is to post a message on our public mailing list at https://accounts.eclipse.org/mailing-list/ospo.zone. You can also come and discuss with us at the usual open source events, join our monthly OSPO OnRamp webinars, or get in touch with any member -- they will kindly redirect you to the right person.

9.2 Appendix: Customised Activity Scorecard template

The latest version of the Customised Activity Scorecard template is available in the `resources` section of the Good Governance Initiative GitLab at OW2.

Goal/Activity Culture 1	**Promote open source development best practices**	**Last update** 07/28/21

Customized Description *Scope of what has to be done*

Brief essential description…

- Brief highlights..
-

Opportunity Assessment *Why is this activity relevant*

- Key pain points…
- Key progress opportunities…

Objectives *What we aim to achieve in this iteration*

- Objective 1…
- Objective 2…

Tools *Technologies, tools and products used in the Activity*

- Resources…

Operational Notes *Approach, method to progress in the Activity*

- Start with…
-

Key Result *How we will measure success in this iteration*	**Progress**	**Score**	**Personal Assessment**
1. Key result 1 (minimum one key result)	**xx%**	**.9**	Personal comment
2. Key result 2	**xx%**	**.5**	Personal comment
3. Key result 3	**xx%**	**.5**	Personal comment
4. Key result 4 (maximum four key results)	**xx%**	**.0**	Personal comment
		.475	

Timeline *Start-End dates, Milestones*

- Date indication here

Efforts *Time and material budget*

- *Time allocation over the next three months*
- *Budget allowance*

Assignees *Who participates? Leads?*

- XX to prepare internal presentation

Issues *Difficulties, uncertainties, roadblocks, points of attention, dependencies*

- Concern 1…
- Concern 2…

Status *How the Activity is doing*

Personal comment on the health of the Activity

Overall Progress Rating	**XX%**

Notes