



WP1

DIGIT B1 - EP Pilot Project 645

***Deliverable 5: General Assessment of the European Institutions and FOSS
Communities' Code Governance Models***

Specific contract n°226 under Framework Contract n° DI/07172 – ABCIII

May 2016



Author:



Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The content, conclusions and recommendations set out in this publication are elaborated in the specific context of the EU – FOSSA project.

The Commission does not guarantee the accuracy of the data included in this study. All representations, warranties, undertakings and guarantees relating to the report are excluded, particularly concerning – but not limited to – the qualities of the assessed projects and products. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use that may be made of the information contained herein.

© European Union, 2016.

Reuse is authorised, without prejudice to the rights of the Commission and of the author(s), provided that the source of the publication is acknowledged. The reuse policy of the European Commission is implemented by a Decision of 12 December 2011.

Contents

CONTENTS	3
LIST OF TABLES	6
LIST OF FIGURES	7
ACRONYMS AND ABBREVIATIONS	8
1 INTRODUCTION	9
1.1. OBJECTIVE OF THIS DOCUMENT AND INTENDED AUDIENCE.....	9
1.2. SCOPE	9
1.3. DOCUMENT STRUCTURE	9
1.4. KEY SUCCESS FACTORS.....	10
1.5. DELIVERABLES	10
2 METHODOLOGICAL APPROACH TO BUILDING THE ANALYSIS	11
2.1. DEFINITION OF THE IMPORTANT GOVERNANCE AREAS TO TAKE INTO ACCOUNT FOR THE ANALYSIS	12
2.2. SELECTION OF EUROPEAN INSTITUTIONS PROJECTS AND FOSS COMMUNITIES	12
2.3. QUESTIONNAIRE DEVELOPMENT AND SURVEYS.....	14
2.4. INFORMATION CLASSIFICATION AND FILTERING PROCESS	14
2.5. ANALYSIS OF THE INFORMATION	15
2.6. DESIGNING A GOVERNANCE MODEL	16
2.6.1. <i>Designing the Governance Model Roles and Responsibilities</i>	16
2.6.2. <i>Defining the Governance Model Processes</i>	18
3 THE GOVERNANCE MODEL	19
3.1. ROLES AND RESPONSIBILITIES	20
3.1.1. <i>Board of Directors/Steering Committee</i>	21
3.1.2. <i>Project Manager/Project Leader</i>	21
3.1.3. <i>Workforce</i>	22
3.1.4. <i>Users</i>	22
3.2. OPEN SOURCE PROJECTS PROCESSES	23
3.2.1. <i>Decision Making Process</i>	23
3.2.2. <i>Support</i>	24
3.2.3. <i>Software Governance Process</i>	24
3.2.4. <i>Financial Support</i>	25

4 ASSESSMENT OF THE EUROPEAN INSTITUTIONS AND FOSS COMMUNITIES' GOVERNANCE MODELS.....27

4.1. ROLES & RESPONSIBILITIES	27
4.1.1. <i>European Institutions</i>	27
4.1.2. <i>FOSS Communities</i>	28
4.1.3. <i>Conclusions and Recommendations</i>	29
<i>Recommendations for European Institutions</i>	29
<i>Recommendations for FOSS Communities</i>	30
4.2. SUPPORT	30
4.2.1. <i>European Institutions</i>	30
4.2.2. <i>FOSS Communities</i>	31
4.2.3. <i>Conclusions and Recommendations</i>	32
<i>Recommendations for European Institutions</i>	33
4.3. DECISION-MAKING.....	33
4.3.1. <i>European Institutions</i>	33
4.3.2. <i>FOSS Communities</i>	34
4.3.3. <i>Conclusions and Recommendations</i>	35
<i>Recommendations for European Institutions</i>	35
<i>Recommendations for FOSS communities</i>	35
4.4. SOFTWARE GOVERNANCE.....	35
4.4.1. <i>European Institutions</i>	36
4.4.2. <i>FOSS Communities</i>	38
4.4.3. <i>Conclusions and Recommendations</i>	40
<i>Recommendations for European Institutions</i>	40
<i>Recommendations for FOSS Communities</i>	40
4.5. FINANCIAL SUPPORT.....	40
4.5.1. <i>Budget</i>	41
4.5.1.1. <i>European Institutions</i>	41
4.5.1.2. <i>FOSS Communities</i>	41
4.5.1.3. <i>Conclusions and Recommendations</i>	41
<i>Recommendations for European Institutions</i>	42
<i>Recommendations for FOSS Communities</i>	42
4.5.2. <i>Funding Model</i>	42
4.5.2.1. <i>European Institutions</i>	42
4.5.2.2. <i>FOSS Communities</i>	42
4.5.2.3. <i>Conclusions and Recommendations</i>	43
<i>Recommendations for European Institutions</i>	45
<i>Recommendations for FOSS Communities</i>	45

5	REFERENCES	46
6	ANNEXES	47
6.1.	ESURVEYS FOR INFORMATION GATHERING.....	47
6.2.	LIST OF RECOMMENDATIONS.....	47

List of Tables

Table 2-1 European Institutions projects selected for the analysis	12
Table 2-2 FOSS Communities selected for the analysis.....	13
Table 4-1 EUI projects status of documentation	30
Table 4-2 EUI projects communication channels.....	31
Table 4-3 FOSS communities status of documentation	31
Table 4-4 FOSS communities communication channels	32
Table 4-5 EUI projects decision making structure.....	33
Table 4-6 FOSS communities decision making structure	34
Table 4-7 EUI projects source code availability	36
Table 4-8 EUI projects definition of requirements	36
Table 4-9 EUI projects software compliance process	37
Table 4-10 EUI projects developers training	37
Table 4-11 FOSS communities source code availability.....	38
Table 4-12 FOSS communities definition of requirements.....	38
Table 4-13 FOSS communities software compliance QA	39
Table 4-14 FOSS communities contributors' training	39

List of Figures

Figure 1. Methodological Approach - Steps.....	11
Figure 2. Methodological approach used to build the analysis - Information sources.....	14
Figure 3. Governance Model	20

Acronyms and Abbreviations

API	Application Programming Interface
EUI	European Institutions
P	European Parliament
ESAPI	Enterprise Security Application Programming Interface
DG	Directorate General
ISACA	Information Systems Audit and Control Association
FOSS	Free and Open Source Software
FOSSA	Free and Open Source Software Auditing
OS	Operating System
OSSWATCH	Open Source Software Watch
OWASP	Open Web Application Security Project
SDLC	System Development Life Cycle
SEO	Search Engine Optimisation
WP	Work Package

1 Introduction

1.1. Objective of this Document and Intended Audience

This document represents the deliverable 5 included within TASK-03: General Assessment of the European Institutions and FOSS Communities' Code Governance Models.

The objective of this document is to explain the fundamental components of a governance model, analyse the governance models and practices used in the European Institutions selected in Deliverable 3, and the FOSS communities selected in Deliverable 4, and provide some recommendations to improve the governance area in general.

This document is addressed to the DIGIT and ITEC departments interested in the results of the governance study and related practices conducted by the European Institutions and FOSS communities. The European Institutions selected projects will benefit from this study as it provides some recommendations to improve their governance area. FOSS communities selected can also take advantage of this deliverable by analysing the recommendations obtained during the study and their applicability within their particular community.

1.2. Scope

This analysis covers the FOSS communities and the European Institutions' projects that are listed in Section 2.2. To conduct this analysis, a representative of the EUI's project or community was invited to complete the survey, published using the EUSurvey [1] application.

Throughout the document, the term 'FOSS communities' refers to the FOSS projects, communities and foundations that fall within defined scope. Red Hat, a private OSS organisation, was included in the analysis at the request of DIGIT. The term 'European Institutions' refers to the European Institutions' projects that fall within the defined scope.

1.3. Document Structure

This document consists of the following sections:

- Section 1: **Introduction**, which describes the objectives of this deliverable, intended audience and Scope.
- Section 2: **Methodological Approach to Building the Analysis**, which describes the steps that we followed to conduct the analysis of the governance models used by the selected European Institutions and FOSS communities, according to the scope.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- Section 3: **The Governance Model.** A governance model is proposed, with a general explanation of each of its components.
- Section 4: **Assessment of the European Institutions and FOSS Communities Governance Models,** which analyses the information gathered from the surveys filled by the selected European Institutions' projects representatives and the surveys filled by everis' team for FOSS communities.
- Section 5: **References.** Main references used in this document.
- Section 6: **Annexes:** Extra documentation related to the study.

1.4. Key Success Factors

All steps described in Section 2 - Methodological approach to building the analysis, will ensure the fulfilment of the key success factors related to this deliverable:

- Best practices include a variety of typologies: technical, organisational and about governance and quality of open source software (e.g.: synchronisation with OSS; guidelines for security software development; secure integration and interoperability of different components; sustainable ways of OSS governance and professional services).

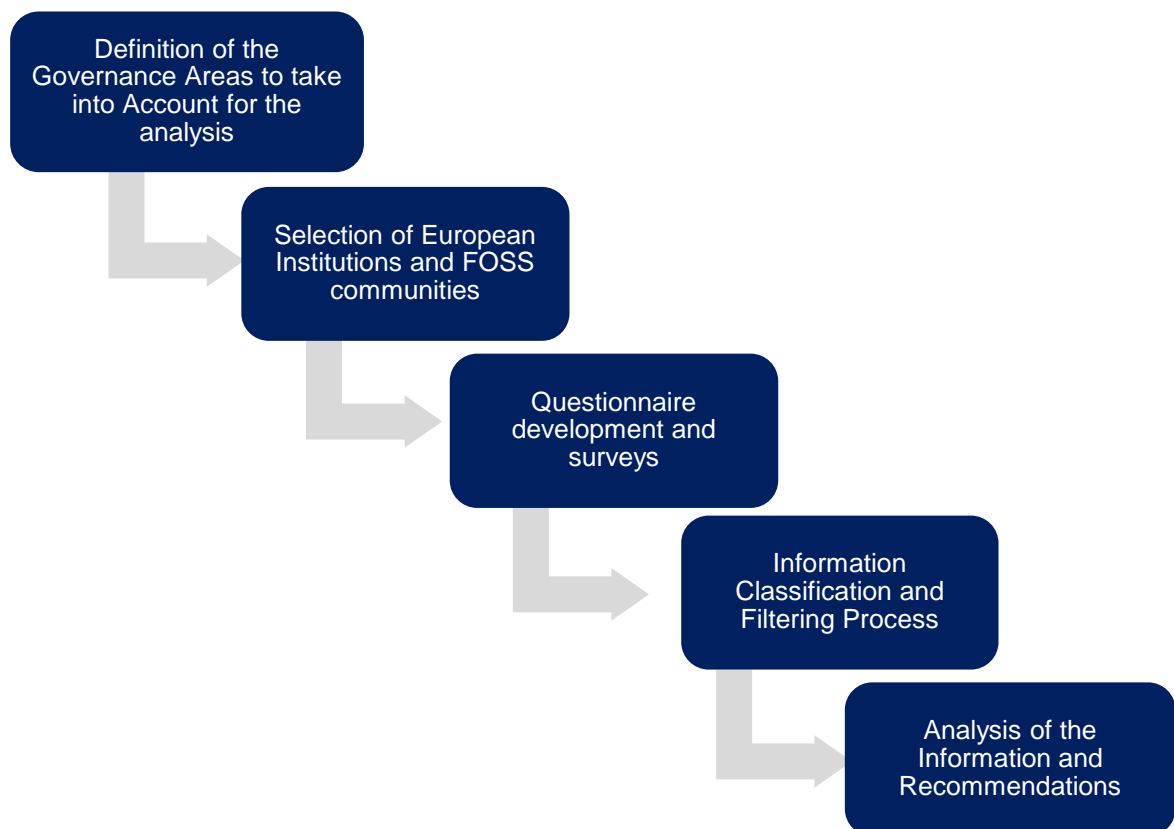
1.5. Deliverables

- 1 *Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions*
- 2 *Deliverable 4 - Analysis of software Development Methodologies Used in the FOSS communities.*

2 Methodological Approach to Building the Analysis

The goal of this document is to analyse all the information gathered during the surveys and research conducted by everis' teams in relation to this study. This analysis will provide valuable information from the European Institutions' projects selected and the identified FOSS communities, with regard to the different Governance areas that we will analyse.

Figure 1. Methodological Approach - Steps



2.1. Definition of the Important Governance Areas to take into Account for the Analysis

Following research on the governance models currently used in open source projects and communities, five areas of governance were defined, as follows:

1. roles and responsibilities
2. support
3. decision making
4. software governance
5. financial support

These code governance areas will be used to build the proposed Governance Model.

2.2. Selection of European Institutions projects and FOSS communities

In order to conduct this analysis, we selected the European Institutions projects and FOSS communities that are depicted in Table 2-1 and Table 2-2.

Table 2-1 European Institutions projects selected for the analysis

No	European Institution	Project Name
1.	European Commission	Project 1
2.		Project 2
3.		Project 3
4.		Project 4
5.		Project 5
6.		Project 6
7.		Project 7
8.		Project 8
9.		Project 9
10.		Project 10
11.		Project 11
12.	European Parliament	Project 12
13.		Project 13
14.		Project 14

Table 2-2 FOSS Communities selected for the analysis

No.	Name of FOSS Community	Description
1.	Apache Tomcat	One of the most popular open source Java Application Servers.
2.	Bitergia	One of the most popular open source software development analytics platforms.
3.	Debian	One of the most famous Linux distributions that only contains open source software.
4.	Drupal	One of the most popular open source Content Management Systems (CMSs) used for websites.
5.	Eclipse	One of the most popular open source IDE (Integrated Development Environment).
6.	Jenkins	One of the most popular open source tools used for continuous integration.
7.	LibreOffice	One of the most popular open source office suites.
8.	OpenSSL	(Core Infrastructure Initiative), one of the most popular toolkits, implementing the Secure Socket Layer (SSL) and Transport Layer Security (TLS).
9.	OpenStack	Open source cloud infrastructure.
10.	OWASP	Open security community.
11.	OwnCloud	One of the most popular open source storage cloud platforms.
12.	Piwik	One of the most popular open source traffic analytics platforms.
13.	Spring	Most widely used Java framework.
14.	Red Hat	A Linux distribution that is sold with commercial support, widely used in enterprise environments.

2.3. Questionnaire Development and Surveys

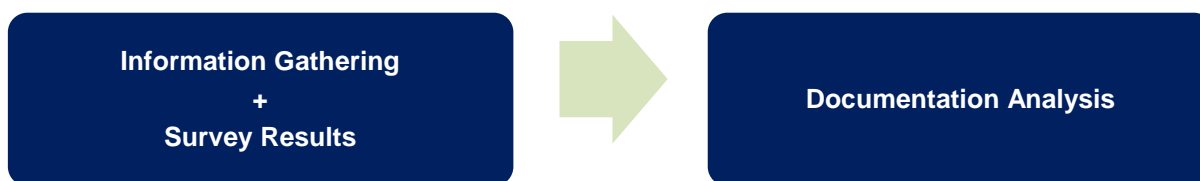
For this step, we conducted the following activities:

- we developed two questionnaires, one for the European Institutions and one for the FOSS communities, with around 18 questions each, and covering all five governance areas defined in step 2.1.
- the questionnaire for the European Institutions was published in EUSurvey [1] and DIGIT requested the participation of the projects' representatives to fill it in.
- the questionnaire for the FOSS communities was published in EUSurvey [1] and an email was sent to the communities' representatives inviting them to fill it in.
- the everis team of FOSS experts provided information on the governance model used by some of the communities.
- the everis project team researched the communities that did not complete the survey, to gather information on their governance models and related practices.
- the everis project team extracted the European Institutions governance information gathered from the surveys.

2.4. Information Classification and Filtering Process

The following figure shows the information sources used to conduct the analysis.

Figure 2. Methodological approach used to build the analysis - Information sources



- **Information Gathering/Survey Results:** For the European Institutions' projects, a questionnaire was published in EUSurvey [1] and an email was sent to the projects' representatives requesting to fill it in. For FOSS communities, a questionnaire was also published in EUSurvey [1] and we invited the communities to fill it in. The information gathered was filtered and classified to conduct the analysis. For this purpose, a spread sheet was created to count the number of projects using a specific governance method, process or practice. Common criteria were taken into account, and the particularities of each community or EUI project were also included, as they could add valuable information to the study.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- **Documentation Analysis:** In order to complete the information related to the identified governance method, process or practice, public documentation found on the communities' websites was analysed to fulfil the governance areas mentioned above. everis' team of experts also provided information for the analysis. After filtering and classifying the data, each governance method, process or practice followed by a community or EUI project was compared with the ones proposed in the governance model and we provided a recommendation, when applicable

2.5. Analysis of the Information

Section 4 of this document is structured as follows:

- **Analysis of identified governance models, processes and best practices used in the European Institutions and FOSS communities:** This section is structured following the five areas of governance identified in Section 2.1.
 - **roles and responsibilities:** It contains the analysis of governance models and practices with regards to organisational aspects such as organisation structure, director boards and committees.
 - **support:** It analyses the practices used for documenting the system and the communication channels used to provide information related to the system, in terms of new features or fixes .
 - **decision-making:** It analyses the methods and practices used to take decisions related to the project future and roadmap, aligned with business goals.
 - **software governance:** It analyses the governance models and practices used for code development governance.
 - **financial support:** It analyses how the projects are funded and how the community or the project ensures its sustainability in the mid and long term.

The usage of each analysed variable is represented by a numeric value and a percentage. To represent these numbers, we used tables to represent the percentage of usage for the total number of projects and communities analysed. It is important to note that the variables are not mutually exclusive; therefore, a project or community can use one or more of them.

To calculate this percentage, we used the following formula:

$$\%usage = nCoincidences * 100 / nProjectsAnalysed$$

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

To view the results of the analysis of the Governance variables for the European Institutions projects, click on the icon:



To view the results of the analysis of the Governance variables for FOSS communities, click on the icon:



2.6. Designing a Governance Model

In order to design a governance model, we analysed the documentation of relevant organisations (see Section 3: The Governance Model), and knowledge gathered during the development of deliverable 3, deliverable 4, and deliverable 6.

Our proposed Governance Model is composed of two parts:

- Roles and Responsibilities, where the contribution of the people is defined.
- Processes, where the main activities related to governance are described.

2.6.1. Designing the Governance Model Roles and Responsibilities

According to the different governance necessities, areas are defined to satisfy those necessities (see Figure 3. Governance Model).

These areas are made up of people with specific roles and responsibilities, to cover the following necessities:

- **Strategy direction:** this group of people decide the strategy for the project or community conduction, take decisions at the organisational level, and deal with other global management matters. This working group is the “**Board of Directors**”.
- **Management:** this group of people manage the current project, and take decisions at the project-level. This necessity will be satisfied by the “**Project Management**” area.
- **Software Development and Maintenance:** Software development projects require several areas to develop and maintain software correctly.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code

Governance Model

- The “**Development**” area develops the software, and is also responsible for bug solving. They also keep up the knowledge base by generating guides of the technologies used.
- The “**QA**” ensures the software quality by inspecting the software and executing tests. It also provides information to improve the way the software is developed, to the corresponding area.
- Software security is ensured by the “**Security**” area, where different security activities are carried out:
 1. security requirements are gathered;
 2. possible threats are analysed;
 3. software is checked to avoid possible vulnerabilities;
 4. documentation is generated for the development area to improve security during software development;
 5. possible development of some critical software and
 6. response to security incidents.
- All of these areas are coordinated by the “**Technical**” area that supervises all the other o, and that takes decisions about software design and software delivery. It also manages other technical necessities of the project or community, like management of the IT infrastructure.
- Moreover, there are some non-technical necessities that could affect the project or community:
 - **Financial sources:** The project or community needs financial sources to keep running; this is managed by the “**Funding Model**” area that looks for ensuring those financial sources.
 - **Budget:** The “**Budget**” area manages the money of the project or community, by providing monetary resources to the different areas and controlling the budget.
 - **Intellectual Property Rights-IPR:** The software developed in the project or community is protected by the Intellectual Property Rights (IPR). The project or community needs to oversee that software contributions are compatible with the software license, and the software license is being followed outside the project or community. This task is carried out by the “**Legal Area**”.
 - **Marketing:** The project or community needs the software to be widely known and used. To achieve this goal, the “**Marketing**” area develops and deploys a strategy that includes promoting the software usage, managing the online channels and attending to forums.
 - **Business:** All of these areas are supervised by the “**Business**” area, that acts as a liaison between management areas and the previous areas. Additional business aspects of the project are managed here too, such as generating documentation of the project or community and relevant processes.
 - **Support:** The project or community needs to aid software users, by means of generating documentation and providing support. This is performed by the “**Support**” area, the main point of contact with the users. This area uses resources both from the “**Technical**” area and the “**Business**” area, to provide technical and non-technical support.

2.6.2. Defining the Governance Model Processes

In the light of the necessities and areas described in the previous subsection, 4 main processes are identified, that affect the project or community governance:

- **Support**, the process that describes how issues, whether technical or non-technical, are solved. It also provides a point of contact of the project or community. This process is critical to maintain software usage, and it has a direct impact on the project or community sustainability.
- **Decision-Making**, the process by which decisions about the project or community are taken. This process should be efficient to avoid delays, and the process should be well documented.
- **Software Governance**, the process of developing and maintaining the software.
- **Financial Support**, the process by which the project or community ensures the continuous flow of monetary resources, and the efficient management of them. This process also has a direct impact on the sustainability of the project or community.

3 The Governance Model

Why does a Project need a governance Model? There are many open source management strategies and open source projects. It is therefore critical that a project clearly communicates how the project is structured and managed. The development and communication of a clear and concise governance model is one of the most important steps a project can take towards its sustainability.

In this document we propose a model based on the necessities analysed in Section 2.6: Designing a Governance Model. The model provides an organisational structure and the main activities that affect governance and sustainability. Moreover, it allows task sharing and efficient management among project and/or community members.

In order for the governance model to be effective, it should follow these guidelines:

1. the governance document should be concise, accessible and easy to refer to;
2. the model should be simple but effective. It should provide a way of guiding the management of the project in a clear and transparent way;
3. the process of third party contributions should be easy therefore adding to the agility of the project;
4. the model should encourage third-party participation by empowering those parties within the project. However, control remains at the top level.

A governance model describes the roles and associated responsibilities that project participants can take on and the relevant processes: decision making, support, software governance and financial support.

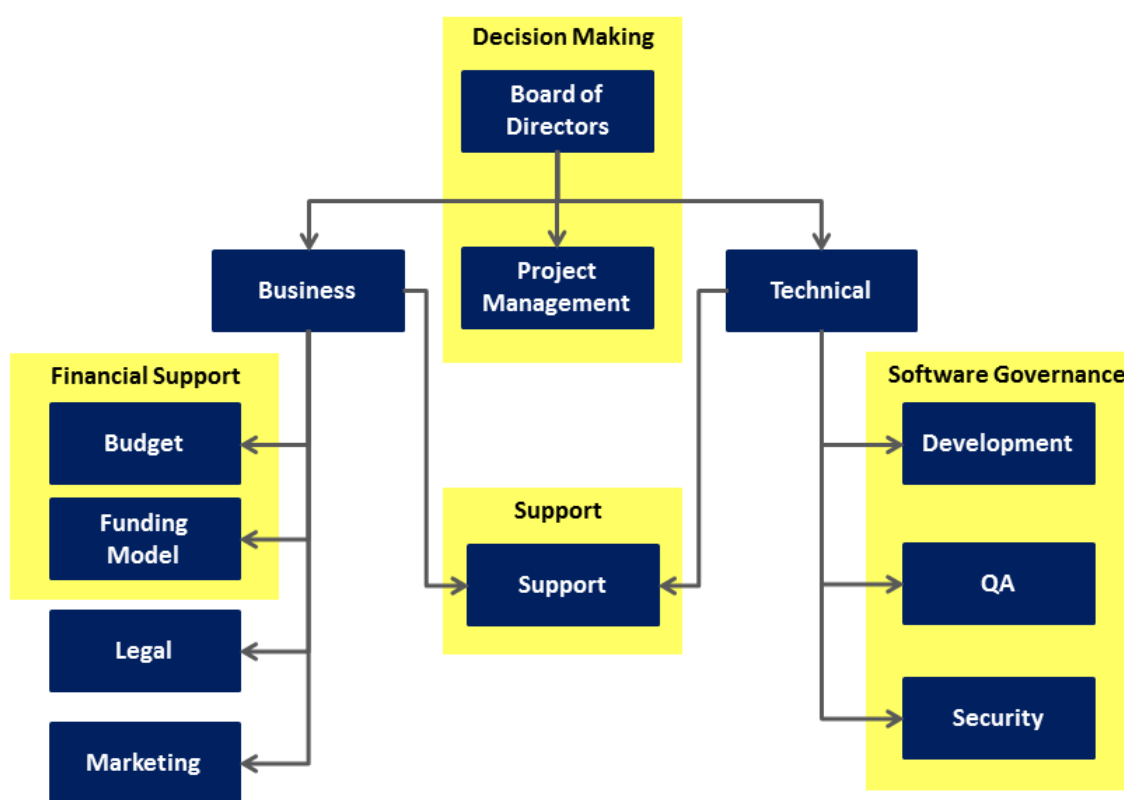
For the proposed model, we researched several reputable and well known organisations, and the websites of the FOSS communities, to gather and analyse their information before designing the model. The most important organisations are:

- **OWASP:** The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organisation focused on improving the security of software. They have a Governance section that explains their roles and responsibilities, their funding method and their decision making approach.
- **OSSWATCH:** Open Source Software Watch is the United Kingdom's advisory service for issues relating to free software and open source software. OSS Watch provides unbiased advice and guidance on the use, development, and licensing of free software, open source software, and open source hardware. They have a Governance section that explains the most frequently used models in the OSS communities, that includes the roles and responsibilities and the processes of decision making, contributions and support.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- **ISACA:** The Information Systems Audit and Control Association is an independent, non-profit, global association, that engages in the development, adoption and use of globally accepted, industry-leading knowledge and practices for information systems. They, together with BlackDuck, analyse the “Open Source Component Governance and Management Using COBIT”.
- Figure 3 depicts the different areas that compose an open source project, highlighting the ones that belong to one of the governance processes identified in Section 3.2 - Open Source Projects Processes

Figure 3. Governance Model



In the following sections each of the proposed governance areas is described in more detail

3.1.Roles and Responsibilities

This section describes the formal roles and their authority within an open source project. The goal of this section is to make it clear who is responsible for decision making, who manages the project, and who contributes.

The roles and responsibilities described here are of a general nature, and each organisation can use them ‘as is’ or create their own to satisfy the needs of the organisation. The important issue is to make sure all the relevant responsibilities are included in each role, to ensure the project’s success.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

In the following section we describe the main roles and responsibilities. Each project or community will decide which ones they find suitable, depending on the size and maturity of the project or organisation.

3.1.1. Board of Directors/Steering Committee

The top management group, which can take different names, i.e. Board of Directors, Steering Committee, is responsible for setting the course of the project and thus ensuring its sustainability.

Their responsibilities include, but are not limited to:

1. set the strategic objectives of the project and communicate these clearly to the project or community;
2. manage internal conflicts;
3. approve changes to the governance model;
4. vote to decide on new member, new staff, etc.

3.1.2. Project Manager/Project Leader

The project leader is responsible for the smooth running of the project, contributing to its overall success.

Their responsibilities include, but are not limited to:

1. prepare and manage the project planning;
2. manage deviations on the project planning and recommend mitigation actions;
3. ensure that the staff make the right decisions on behalf of the project. Generally speaking, as long as the staff group is aligned with the project's strategy, the project lead will allow them to proceed as they desire;
4. review code contributions;
5. manage the copyrights within the project outputs;
6. manage the access to the project's private mailing list and its archives. This list is used for sensitive issues, such as votes for new members and legal matters that cannot be discussed in public. It is never used for project management or planning;
7. assign access rights to staff;
8. ensure that communication mechanisms exist to ensure that all contributions are reviewed by the community as a whole.

3.1.3. Workforce

Workforce comprises all the people that are responsible for covering all the tasks defined for each of the governance processes within the project or community. They constitute the engine of the project or community. They perform technical and non-technical tasks, depending on the process they belong to.

Workforce can be assigned to different working groups, depending on the size of the project and their maturity. However, all of the following activities should be accomplished one way or another:

Technical tasks:

- provide technical support for users;
- report bugs;
- identify requirements: business, technical, security;
- provide graphics and web design;
- Conduct software development;
- assist with project infrastructure;
- write documentation;
- fix bugs;
- add new features or improve existing ones;
- test the software.

Non-technical tasks:

- provide non-technical support for users;
- promote the software under development and other marketing actions, like event attendance and media channels, to increase the engagement of the users and aid in the sustainability of the project;
- look after software license and Intellectual Property Rights (IPR) of the projects;
- manage the project or community budget and sponsorship.

3.1.4. Users

Users are members of a community or project who have an interest in the project. Their importance lies in the fact that without them the project would have no purpose.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

The project needs the users' involvement in the project and/or community as much as possible, as their participation will ensure that the project is satisfying their needs.

Users who continue to engage with the project and its community will often become more and more involved. Such users may find themselves becoming contributors.

Some of the user responsibilities are (but are not limited to):

1. evangelising about the project (e.g. a link on a website and word-of-mouth awareness raising);
2. informing developers of strengths and weaknesses from a new user perspective;
3. providing moral support, by acknowledging the project's positive points;
4. providing financial support.

3.2. Open Source Projects Processes

3.2.1. Decision Making Process

Decision making can be a complex issue, and it is difficult for a committee to reach a decision on an open source project issue. However, the way in which decisions are made is critical to the success of an open source software development project. A process that is too time consuming may result in decisions being deferred, thus delaying the project. Conversely, a process that is not well defined may result in arguments about whether a decision is valid or not, again resulting in delays and non-agreements within the project members.

A decision making process should be documented, taking into account the following:

1. the process should be transparent, and as such, final decisions should be communicated to all of the relevant members;
2. conflict resolution must be part of the process;
3. decisions about the future of the project are made through discussion with all members of the community, from the newest user to the most experienced PM member;
4. decision making for all other issues is done through discussions with the management group, and relevant experienced members of the project or community, and it typically involves the following steps:
 - a. proposal of the new idea;
 - b. discussion of the idea;
 - c. If consensus is not reached through discussion, then voting is required. Voting is usually done for issues such as those affecting the strategic direction or legal standing of the project.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

d. final decision.

5. the voting process should be clearly described, so as to not leave room for interpretations.

3.2.2. Support

For most open source projects, the support channels are the main contact with the users of the project. These users may become future staff or customers of commercial support providers, and are therefore key to the sustainability of a project. An open source project must look after its users.

This section also describes the status of documentation, which is very important for the quality of the project deliverables. Documentation provides users with a roadmap of the system, and technical staff with the guidelines for development of new features and support of the existing ones.

All participants in the community are encouraged to provide support for new users within the project management infrastructure. This support is provided as a way of growing the community. Those seeking support should recognise that all support activity within the project is voluntary and is therefore provided as and when time allows. A user requiring guaranteed response times or results should therefore seek to purchase a support contract from a project. However, for those willing to engage with the project on its own terms, and willing to help support other users, the community support channels are ideal.

3.2.3. Software Governance Process

As far as software governance is concerned, several variables provide information about how the software development is managed. These variables are:

- source code availability: This variable aims to provide information about the confidentiality among software development teams, and the visibility of the software development among the software development teams.
- software requirements definition: This variable measures how the software objectives are defined, as well as the standardisation of the process.
- quality assurance - QA: This variable aims to analyse how the software quality is checked, as well as the compliance with code conventions and best practices. This area also incorporates compliance processes to ensure the software quality. QA staff will check if the software satisfies the requirements defined.
- training: This variable provides information about how the developers are trained within the development teams for the execution of their tasks. This learning material can be composed of:
 - specific knowledge tailored to the current development;
 - basic knowledge of used technologies;

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- organisation of the software development team, and its processes.

It is important to note that all the software development tasks should comply with the software requirements definition to ensure they satisfy the user's needs. All of these tasks will also be validated during Quality Assurance. These tasks are covered in detail in *Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions* and *Deliverable 4 - Analysis of software Development Methodologies Used in the FOSS communities*.

3.2.4. Financial Support

Both European Institutions and FOSS communities require some type of financial support. Without it, they cannot survive as they need funds to cover the expenses related to the project. These funds can take the form of a formal budget or some type of sponsorship.

European Institutions have to be backed by a budget to cover the expenses related to the project, and as such, they rely on an assigned budget that is distributed to all the project related tasks.

FOSS communities, even though they are supported by a lot of volunteers or contributors, need funds to pay the external services or infrastructure they need. Thus, it is important to have an income to ensure the sustainability of the community.

Financial support can be provided in two ways

1. Budget:

Budget is created to control and decide which expenses are assigned to which part of the project in order to maintain their sustainability. This ensures that an estimation of resources will be allocated for the budget's period.

Budget should include:

1. maintenance and development of new features;
2. staff, for technical and non-technical tasks;
3. project support;
4. infrastructure;
5. training, seminars and workshops;
6. branding / marketing.

2. Funding model:

Funding can be done in several ways, among them:

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

Sponsorship: Sponsorship might be necessary to receive external funds for the project that could be used to assign specialized contributors.

Sponsorship is frequently supported by the fact that the sponsor will benefit from the project, for several reasons:

1. they are using either totally or partially the FOSS application/library for their projects;
2. they are receiving some type of service or because they use the software for their own benefit;
3. for branding and marketing purposes, organisations can use the brand of the community as a way to improve their own image.

Sponsorship does not necessarily entitle a monetary provision; it can be through the organisation's internal channels supporting the project, and can take the form of a provision of infrastructure services, branding of the community or project, etc.

Donations: Donations from companies, public organisations and individuals are welcomed by FOSS communities, as they provide the much needed funds to pay for external services or infrastructures.

For European Institutions, this does not apply

Events: Another way to receive funds are conferences. Companies, universities and institutions organise conferences where communities' representatives are contracted as speakers.

4 Assessment of the European Institutions and FOSS Communities' Governance Models

4.1.Roles & Responsibilities

In the following sections we provide an analysis on how the roles and responsibilities are defined in both European Institutions and FOSS communities

4.1.1.European Institutions

The roles are clearly defined among European Institutions' projects, where not all governance areas are in the projects. Some areas are usually managed by parties outside the project, but the activities of those areas can be conducted also within the project if needed. Some of the areas external to the project are:

- legal area, which is managed within the European Institutions. The responsibilities of this area are the management of software license and Intellectual Property Rights;
- marketing area, which is managed by other parts of the European Institutions;
- security area, which is managed by other parts of the European Institutions (DIGIT, CERT-EU, etc.);
- business area, which is usually managed by the project owner;
- as far as the Sponsorship area is concerned, European Institutions are the sponsors of their own projects.

The areas that are managed within the project include the following:

- the strategic direction and decision making, in which the Board of Directors is usually composed of committees like the steering committee, and they collaborate in the decision-making process;
- in European Institutions', Project Management activities are conducted by the project manager of the project;
- technical area, comprised of technical project leaders like the software architects of the project;
- development, which is made up of staff that develop the software of the project;

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- QA area, which is composed of software testers within the project support, which is made up of staff that provides support of the project. This area can also have staff outside the project, especially for the first support level (helpdesk). Upper levels like specialized support, have usually staff from the project.

In European Institutions, we do not have information on the specific roles and responsibilities that are defined for each area.

4.1.2. FOSS Communities

FOSS communities are usually organised into working groups/teams that are in charge of specific tasks. These tasks are focused on some specific area, such as the development of a software component, a security module, etc. The number of these groups is often related with the size of the community, where bigger communities usually have a larger number of groups.

Some group members lead the group by means of managing the task execution and group resources. These leaders are mostly in contact with management community groups, being the liaison between decision-making groups and working groups.

The decision-making groups are usually management boards or committees, where core community members manage the strategy of the community. In those communities that have different sponsors, most of the management groups are formed by sponsors' staff that take care of sponsors' interests and ensure vendor-agnostic community position.

When a community is made up of different projects, there are project leaders that manage the project. In this context, those project leaders are the point of contact between management groups and working teams.

In FOSS communities, there are roles that are well defined, like the following:

- board of directors and project management staff (explained in the paragraphs above);
- technical staff, community members with technical skills and high level of commitment with the community, who also coordinate documentation development, answer FAQs and provide technical support;
- development staff, contributors that develop software for the FOSS;
- QA staff, contributors that review the software contributions of other community members to ensure the quality of software and ensure compliance with code conventions;
- security staff, they deal with possible security issues;

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- support staff, they provide support for FOSS users and community newcomers. Support activities in FOSS communities include documentation development and the management of FAQs and community forums;
- budget and sponsorship staff, people in charge of managing the financial support of the community, and the relationships with the community sponsors. This role is of utmost importance as it has a direct effect on the community's sustainability.

Moreover, other roles related to the business aspect of FOSS communities are defined, especially in big communities, among them:

- legal staff, community members that are in charge of looking after the licensing of FOSS, as well as the legal aspects of the contributions. In small communities, this role is not clearly defined;
- marketing and sponsorship staff, community members that are in charge of fostering FOSS usage, improving attractiveness for possible contributors, and managing the community's social network.

For those communities created by private organisations, several roles are usually assigned to the private organisation staff, mostly supervisory and decision-making roles.

4.1.3. Conclusions and Recommendations

In the light of the previous information Roles and Responsibilities are, in general, aligned in European Institutions and FOSS communities in terms of the organisation. Roles and Responsibilities are distributed in a similar way in the project or community. However, there are a few intrinsic differences, as a result of their nature:

- some governance areas in the EUIs' projects are conducted by staff outside the project;
- FOSS communities can have several sponsors while EUIs' projects only have one;
- management and decision-making roles are quite diverse among FOSS projects, while European Institutions usually have the same base structure.

Recommendations for European Institutions

- ensure that all the roles and responsibilities, whether technical or business-related are clearly defined and documented.
- ensure the existence of a Steering Committee/Board of Directors charter document.

Recommendations for FOSS Communities

- ensure the existence of a Steering Committee/Board of Director charter document;
- clearly define all the business-related roles and responsibilities (legal area, marketing area, etc.), for all FOSS communities, with emphasis in those that can directly impact the sustainability of the community, like marketing and sponsorship

4.2.Support

Support covers the areas of system documentation and communication channels with end-users to provide information on new features added to the system, new releases and bug fixes.

It is important to mention that all the support processes and tasks related to the System Development Lifecycle are analysed in *Deliverable 4 - Analysis of software Development Methodologies Used in the FOSS communities*, and therefore are not part of the governance area. However, their compliance is evaluated during Quality Assurance

4.2.1.European Institutions

As far as European Institutions are concerned, support activities are addressed to provide software users and system administrators of the European Institutions with detailed information in the form of documentation. Multiple channels can be used to mitigate and solve end-user problems.

Table 4-1 EUI projects status of documentation

Status of Documentation	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Updated	1	8%
Partially Updated	8	62%
Not maintained	2	15%
Updated by externals	0	0%
N/A	2	15%

Most of the European Institutions analysed projects include documentation about their software, but it is only partially updated. Only a minority of them keep the documentation up-to-date, and a few of them (about 15%) do not maintain their documentation.

Table 4-2 EUI projects communication channels

Communication Channels	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Yes, official channel is established	12	92%
No-official channel, but there is communication with users	1	8%
N/A	0	0%

While 8% of the projects do not have an official channel to communicate with the users, mainly because it is an obsolete system, the remaining projects have official communication channels with end-users to announce new features of the system, new releases and bug fixes. These channels include Yammer, email, Helpdesk, phone, website, open door sessions, wiki and Intranet among others.

4.2.2. FOSS Communities

For FOSS communities, support activities are quite important in terms of community sustainability. The usage of FOSS depends on the support that the community provides. If the FOSS is used frequently, the sustainability of the community is improved.

Table 4-3 FOSS communities status of documentation

Status of Documentation	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Updated	12	86%
Partially updated	2	14%
Not maintained	0	0%
No documentation	0	0%

Most of the communities have updated the documentation of their software in order to ease the FOSS utilisation. This documentation is mostly composed of installation guides, configuration guides and FAQ forums or documents. Once a new version of FOSS is planned for release, all related documentation is generated or updated, as needed.

Table 4-4 FOSS communities communication channels

Communication Channels	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Yes, official channel is established	13	93%
No-official channel, but there is communication with users	1	7%
No	0	0%

For FOSS communities, the communication with FOSS users is an important variable to take into account. By means of communicating with FOSS users they adapt the FOSS to users' needs and mitigate or solve the possible bugs detected during the system operation. That is why most of the FOSS communities have a plethora of communication channels, such as wikis, websites, email lists, etc.

These channels are used to provide users with information about FOSS installation, FOSS utilisation, and FAQs. Furthermore, users can provide their comments about possible improvements, doubts or any other piece of relevant information.

4.2.3. Conclusions and Recommendations

Keeping the documentation up-to-date is a good way to improve the support effectiveness. Users can get the information that they require without consuming additional support resources. This results in an increase of the efficiency of user support activities, and/or the reduction of the efforts required for providing those activities.

Moreover, a well-documented software is more attractive to users. That is why FOSS communities dedicate efforts to keep the documentation updated, since FOSS usage affects the overall community sustainability.

As for the official communication channels, FOSS communities and most of the European Institutions' projects have official channels to exchange information with end-users. These channels are different in European Institutions and FOSS communities due to their nature. FOSS communities use online channels to provide support to users, such as emails and forums. European Institutions also have online channels, as well as Helpdesk.

Recommendations for European Institutions

- keep the project documentation up to date at all times so as to help end-users and system administrators.

4.3. Decision-Making

The decision making process is a complex one and it should be supported by some type of organisational structure, whether formal or informal. This requires guidelines or procedures to guide the decision process, ensuring that decisions are taken in a timely and objective way, avoiding unnecessary delays and arguments.

4.3.1. European Institutions

Table 4-5 EUI projects decision making structure

Decision Making Structure	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Steering Committee	8	62%
PMO Related	4	31%
Technical Committee	4	31%
No Structure	2	15%

All the European Institutions' projects have a project management team that takes decisions about the future of the project. Furthermore, most of the analysed projects have a steering committee that is part of the decision-making process. Some of the projects (about 31%) have a PMO-related or technical committee that takes part in the project decision-making process.

The majority of the analysed projects are consistent with business goals and existing roadmaps. However, some of them (about 15%) are not, mainly due to the nature of the project or obsolescence of the current system (in this case the new system will be aligned with business goals).

4.3.2. FOSS Communities

Table 4-6 FOSS communities decision making structure

Decision Making Structure	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Committees (Steering or others)	8	57%
Benevolent Dictatorship	1	7%
Private Organisation	5	36%

FOSS communities usually have clearly defined management entities to take decisions about the community. These entities can be:

- **benevolent dictatorship:** It is made up of a small number of leaders that are directly related to the open source project. Typically they are the project founders, who are responsible for defining the general objectives and making the final decisions related to these projects. Usually this position is permanent, which means that the leadership remains unchanged.
- **committees:** Most of the analysed communities use management groups/teams where community leaders take decisions about the community. These teams or committees are elected periodically by community members. So as to be eligible for the committee seats, community members have to demonstrate their commitment by means of merits.
- **private organisation:** When a FOSS community is created by a private organisation, this organisation usually takes all community-related decisions, and it decides about the future of the community. It might be similar to the Benevolent Dictatorship, as a single leader (in this case a private organisation) takes decisions on behalf of the community. Usually this private organisation is the founder of the community.

Due to the nature of FOSS communities, FOSS projects are aligned with the objectives and goals of the community. FOSS communities are created with a clear objective, namely, to satisfy users' software needs following the 'open' philosophy. The projects within a community are aligned with the community objectives, even though FOSS projects can have some degree of independence. FOSS project leaders are usually part of the management committees, and they are part of the decision-making teams.

4.3.3. Conclusions and Recommendations

European Institutions' projects are quite homogeneous in terms of project organisation, where the project management area usually takes the decisions, supported by additional areas.

On the other hand, FOSS communities are heterogeneous in terms of project organisation. They are organised in committees, and have different groups in charge of different areas of the community. One of them is the board of directors or a similar group, in charge of the decisions concerning the future of the community.

As far as the consistency with goals and roadmaps is concerned, FOSS communities are aligned with goals and existing roadmaps, whereas some European Institutions' projects are not aligned but have a justification for not doing so.

Thus taking all this information into account, the following recommendations are provided:

Recommendations for European Institutions

- be aligned with the roadmap and business objectives, for all projects;
- document the decision making process, following the guidelines provided in section 3.2.1 of the Governance Model;
- comply with the approved decision making process.

Recommendations for FOSS communities

- document the decision making process, following the guidelines provided in section 3.2.1 of the Governance Model;
- comply with the approved decision making process.

4.4. Software Governance

Software governance encompasses the transparency of the project in the form of availability of the source code, the clear and concise definition of the requirements whether functional, technical or security related, the quality assurance process and the necessary training of the most relevant people.

4.4.1. European Institutions

Table 4-7 EUI projects source code availability

Source Code Available	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Yes	11	85%
No	1	8%
N/A	1	8%

In terms of source code availability for all developers, most of the analysed EUIs' projects make it available, but 8% of them do not because of the nature of the project. Source code availability allows improving the understanding of the software under development. By having a clear view of other development teams, the integration of different software components is easier because the code can be checked.

Table 4-8 EUI projects definition of requirements

Definition of Requirements	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Divided in business and functional requirements	10	77%
Not divided	1	8%
N/A	2	15%

Most of the analysed projects divide the software requirements according to their nature. This is an evidence of the formality of the requirement gathering process. If the requirements are divided into categories, this process is standardised and the requirement gathering task is done in an orderly way.

Table 4-9 EUI projects software compliance process

Software Compliance Process	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Formal	7	54%
Public or Informal	1	8%
No existing process	3	23%
N/A	2	15%

More than half of the analysed projects (about 55%) have a formal approach to compliance, so as to ensure the quality of the software developed. Some of the analysed projects (about 23%) do not contemplate or do not know of a process to ensure the quality of their software.

Table 4-10 EUI projects developers training

Developers' Training	Out of the 13 Analysed Projects	
	Number of Projects	Percentage of the Projects
Yes	11	85%
No	1	8%
N/A	1	8%

Most of the analysed projects have a training program for developers, so as to improve and maintain their knowledge and capacities within the project. Only 8% of the analysed projects do not provide any training for developers, mainly because of the nature of the project itself. Training developers improve development time and software quality, apart from keeping the knowledge within the project.

4.4.2. FOSS Communities

Table 4-11 FOSS communities source code availability

Source Code Available	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Yes	13	93%
No	0	0%
N/A	1	7%

As it can be seen in the previous table, most of the FOSS communities have made their source code available to all software developers. This is logical, given the nature of FOSS, and also to allow any contributor to understand all of the software components and any dependencies that may exist.

Table 4-12 FOSS communities definition of requirements

Definition of Requirements	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Yes, divided in functional ones, business information and technical ones	4	29%
Yes, not separated	3	21%
No	0	0%
N/A	7	50%

The definition of requirements in FOSS communities is divided into two groups: the FOSS communities that define requirements according to their nature, and those that define requirements without classifying them. Requirement classification is a measurement of the maturity of the requirement gathering process.

Table 4-13 FOSS communities software compliance QA

Compliance Process for Contributions	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Yes, formally defined	11	79%
Yes, no formally defined	1	7%
No	0	0%
N/A	2	14%

So as to ensure the good quality of the FOSS, most FOSS communities have a compliance process for contributions. This process aims to guarantee the software quality and security. This process also impacts support activities, since bad software will require more support, efforts and resources.

Table 4-14 FOSS communities contributors' training

Contributors' Training	Out of the 14 Analysed Communities	
	Number of Communities	Percentage of the Communities
Tailored to current version of the software components used	11	79%
Basic Training of base Technologies	3	21%
Explanation about the organisation and related processes	12	86%
No	1	7%
N/A	1	7%

According to the results of the previous table, most of the FOSS communities provide training tailored to the current development needs. The minority of them provide training for basic technologies that are used in the software development. Additionally, most of the FOSS communities provide information about the community's organisation and processes.

4.4.3. Conclusions and Recommendations

In terms of source code availability, most of the analysed communities and EUIs' projects are aligned in the way they made their code available to developers. They are aligned in this aspect, but it's worth noting that not all European Institutions' projects do it. And this is justified by the nature of the project.

Regarding the definition of software requirements, most of the European Institutions projects gather the requirements and classify them into categories (functional, business information and technical). In contrast, only a minority of FOSS communities do this. This could be a result of lack of information on the QMS system

While most of the FOSS communities have standardised and defined compliance processes for software contributions, it is only implemented in over half of the EUIs' projects. This is quite important to ensure the quality of software, and all EUIs' projects should implement it.

FOSS communities and EUIs' projects are aligned in terms of training for developers. Most of the analysed communities and EUIs' projects provide training materials for developers so as to improve the knowledge within the community/project, or to maintain the knowledge with regard to current technologies.

In view of the previous information, the following recommendations are provided:

Recommendations for European Institutions

- share the code among the projects;
- define software requirements, and classify them according to their nature, i.e. business, functional, technical;
- have a formal and defined compliance process for software contribution.

Recommendations for FOSS Communities

- define software requirements, and classify them according to their nature, i.e. business, functional, technical.

4.5. Financial Support

Financial support is a mandatory requirement for the completion of a project, its future success and to ensure the mid and long term sustainability.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

A project needs funds to cover the expenses related to its development and maintenance, and the sustainability of the community.

4.5.1. Budget

Budget is created to control and provision the expenses for the project in order to maintain their sustainability. This ensures that an estimation of resources will be allocated for the budget's period.

4.5.1.1. European Institutions

Most of the analysed projects from the European Institutions have an annual budget for the maintenance and development of the applications. The areas covered by this budget include:

- external services: external consultants, mostly working on-site, who develop and maintain the projects;
- provider's support: some projects pay the professional services and support of the application's providers;
- infrastructure: technological infrastructure and related services to support the applications.

4.5.1.2. FOSS Communities

Most of the analysed communities rely on an annual budget, which is public for most of the projects. The main expense covered by this budget is the employees, as some communities hire people to work on specific areas of the community.

4.5.1.3. Conclusions and Recommendations

Most of the budget of the European Institutions projects is assigned to development and infrastructure expenses, this is not the case for FOSS communities. Since most of the professional and development services are provided by contributors or volunteers who take advantage of existing tools or infrastructure (which are free for open source projects), most of the budget goes towards the payment of seminars, branding and workshops to promote or spread what the community is doing. Budget is also used to pay the professional services of their employees. We have to consider that European Institutions uses additional budgets to pay for management, branding and other services.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

The efforts made by open source communities to conduct seminars and workshops everywhere is appropriate in terms of social benefits, since they share the knowledge with everyone interested. This is also a good strategy for branding, networking and attracting the most talented people.

To sum it up, it is a good practice to have a budget that will provide a realistic view of what the expenses will be for the specified period, so project resources can be allocated in advance.

Recommendations for European Institutions

- have a realistic annual budget;
- ensure the funding resources allocation for the whole project lifecycle;
- engage project owners to ensure budget provision for the years ahead.

Recommendations for FOSS Communities

- have a realistic annual budget.

4.5.2. Funding Model

Even though FOSS communities are supported by a lot of volunteers or contributors, they still need funds to pay for the external services or infrastructure they need. Thus, it is important to have an income to ensure the sustainability of the community, which can take the form of a formal budget or some type of funding that will help achieve the project's goals. European Institutions' projects, although backed by a budget, need to also ensure that the project will be supported in the future, and as such should consider some type of non-monetary funding.

4.5.2.1. European Institutions

Most of the analysed projects do not receive external funds, the project budget covers all the expenses related to the project development and services needed. Only one project receives external funds from other European Institutions when they use the services provided by the project.

4.5.2.2. FOSS Communities

In this point we analyse the different funding models used by the analysed projects.

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- some FOSS communities receive monetary funds from sponsors, normally these sponsors have motivations towards the applications or services provided by the community, among them are:
 - they are using the whole or some parts of the FOSS application / library for their projects;
 - for branding and marketing purposes, organisations can use the brand of the community as a way to improve their image.
- another way to receive funds is from conferences. Companies, universities and institutions organise conferences where communities' representatives are contracted as speakers;
- funds can also come from donations from private companies, public organisations and individuals.

4.5.2.3. Conclusions and Recommendations

There are some similarities between the funding approaches followed by EUI and FOSS communities, but there are a lot of differences, since their goals are different.

European Institutions have a budget that covers the necessary resources for the project, mainly payment of professional services. Most of the applications developed by the European Commission are not open source, and they are developed to satisfy the requirements of a specific sponsor. Some communities follow a similar approach, as they are created as commercial companies which use open source products, offering a professional services subscription (Red Hat) or re-licensing (Apple Inc.).

Some European Institutions' projects receive funding from other EUI, who in turn can use the software or services.

FOSS communities use different approaches to fund their projects, such as:

- dual-Mode Licenses: Some projects use this approach, which combines the usage of two types of licenses. One is free and authorises the basic use of the application, paying for enabling extra features or enhancements;
- branding: Some communities sell POP material, such as T-Shirts, stickers and mugs with the logo or brand of the community;
- enterprise services: These services offered by the community include training, technical support or consulting. They also provide physical support, like CD/DVD, or complete installations, like executables or pre-configured virtual machines;

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- professional services Subscriptions: Some projects, such as Red Hat, fork the community version, which is free, and offer professional services subscriptions with extra features and services;
- SaaS Services: Some organisations offer their projects in SaaS mode, which consist in subscriptions to the software offered online. The customer does not to invest in a complex IT infrastructure and installation;
- franchising: Some projects use this innovative approach, where commercial partners can use the brand of the company and its logo. **Moodle** is a good example of adoption of this approach;
- donations: Some projects are funded by donations made mostly by individuals, public organisations or companies that use software which is useful for them and in return they want to collaborate for its sustainability. There is a discussion about this model because it seems that it is not a suitable approach to ensure the sustainability of some projects;
- crowd funding: It is a model inspired in the pre-paid approach. There are specialized web sites, such as kick-starter, where projects are posted and explained, allowing people interested in the product the opportunity to pay an amount in exchange of some goods. This is not really suitable for Open Source projects since most of the time is difficult to calculate the effort to create;
- partnership: Some organisations are interested in the product and they collaborate with the community sharing some resources, like infrastructure, programmers or events creation;
- advertising: Some projects or web sites allow the inclusion of publicity on their interface, this approach allows exploiting an ad-based business model. One of the most successful companies that uses this approach is Google;
- sponsorship: Some projects are funded by sponsors' who are interested in them;
- bounty-driven: When some end users or organisations are interested in a specific feature, they can pool money and the community will start to develop the feature requested;
- proprietary parts: Some projects offer the software as open source, but if you want to use the application as a whole you have to buy some proprietary parts like music or graphics;
- re-licensing: Some less restrictive licenses allow for re-licensing the product and sell it as a different one such as Apple and BSD;
- source code obfuscation: It is a discussed approach by the open source community, it consist to obfuscate the crucial parts of the software and offer the rest as a normal open source product. Free Software Foundation is clearly against this practice;

Deliverable 5: General Assessment of the European Institutions and FOSS Communities Code Governance Model

- events organisation: Some communities receive funds organising events such as conferences and workshops;
- Investments: Common in start-ups, they expose the project to some investors in order to receive funds to develop the project. Some FOSS projects have been initiated using this approach;
- Version Lagging: Some projects provide the latest version or update only for subscribers. After a period of time or a new version release, the old version becomes open source;
- Free for Open Source: Some communities provide their software as free for open source projects, meaning that projects which are commercial or proprietary in nature have to pay the license of the product.

As it is showed in this section, FOSS communities have to discover and innovate to find ways to ensure their sustainability. Most of the FOSS communities use a combination of the approaches aforementioned.

Some of the interviewees declare that there are some FOSS projects which are widely used but do not receive enough support. This is because contributors and sponsors are interested in new and visible projects. It is the case for some open source security libraries which are widely used and ensure the privacy of communications. One interviewee mentioned that these kind of projects should be supported by governments or public institutions to protect the security and confidentiality of data and IT communications of the users, as they ensure the security in roads or railways.

Recommendations for European Institutions

- set up a win-win approach for funding open sources projects which are critical for the sustainability of the EUI projects or even beneficial for the society.

Recommendations for FOSS Communities

- diversify funding approaches to engage the major number of sponsors and users willing to use the services, products or extra features;
- create alliances with public and private organisations for resources sharing;
- offer value-added services such as support, training and consulting.

5 References

[1] European Commission, “EUSurvey,” [Online]. Available: <https://ec.europa.eu/eusurvey/home/welcome>.

1. <https://www.blackducksoftware.com/resources/open-source-terms/open-source-governance#sthash.wOivhz00.dpuf>
2. <http://oss-watch.ac.uk/resources/governancemodels>
3. <http://oss-watch.ac.uk/resources/benevolentdictatorgovernancemodel>
4. <http://oss-watch.ac.uk/resources/meritocraticgovernancemodel>
5. <http://oss-watch.ac.uk/resources/rolesinopensource>
6. <http://www.slideshare.net/outercurve/2013-0508-governance-models>
7. <https://www.isaca.org/Education/Online-Learning/Documents/OSS-Government-Management-using-COBIT-5.pdf>
8. <http://timreview.ca/article/636>
9. http://conferences.sun.ac.za/public/conferences/25/slides/IT-Section-2015-Martilene_Orffer.pdf
10. <https://opensource.com/business/14/4/governance-openstack>
11. <https://wiki.openstack.org/wiki/ProductTeam#Methodology>
12. <https://eclipse.org/org/documents/>
13. <https://www.eclipse.org/org/workinggroups/>
14. <https://www.owasp.org/index.php/Governance>
15. <https://www.libreoffice.org/about-us/governance/>
16. <http://www.apache.org/foundation/how-it-works.html#meritocracy>
17. <https://www.openssl.org/policies/>
18. <https://fedoraproject.org/wiki/Council#Introduction>
19. <https://owncloud.org/privacy/>

6 Annexes

6.1.eSurveys for Information Gathering

To view the Governance questionnaire for the European Institutions, double click on the icon:



Governance
Questionnaire-EU Insti

To view the Governance questionnaire for the FOSS communities , double click on the icon:



Governance
questionnaire-FOSS-c

6.2.List of Recommendations



Governance
Recommendations