

WP1

DIGIT B1 - EP Pilot Project 645

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

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

January 2016

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

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.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Contents

| CONT | ENTS |
|--|---|
| LIST O | F TABLES |
| LIST O | F FIGURES |
| ACRO | NYMS AND ABBREVIATIONS10 |
| 1. IN | RODUCTION11 |
| 1.1. | OBJECTIVE OF THIS DOCUMENT AND INTENDED AUDIENCE |
| 1.2. | SCOPE |
| 1.3. | DOCUMENT STRUCTURE |
| 1.4. | KEY SUCCESS FACTORS |
| 2. ME | THODOLOGICAL APPROACH TO BUILDING THE ANALYSIS |
| 2.1. | Selection of Projects, Engagement with the European Institutions and Information |
| GATH | IERING13 |
| 2.2. | : INFORMATION CLASSIFICATION AND FILTERING PROCESS |
| 2.3. | ANALYSIS OF THE INFORMATION |
| | |
| LIBRA | FTWARE DEVELOPMENT METHODOLOGIES, BEST PRACTICES, FRAMEWORKS, RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN |
| LIBRA INSTIT 3.1. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC 3.3. 3.4. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC 3.3. 3.4. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC 3.3. 3.4. DEVE 3.5. | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC 3.3. 3.4. DEVE 3.5. 4. AN | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN 16 METHODOLOGIES USED BY THE ANALYSED PROJECTS DURING THE SOFTWARE DEVELOPMENT 17 16 BEST PRACTICES USED BY THE ANALYSED PROJECTS DURING THE SOFTWARE DEVELOPMENT 17 16 16 16 BEST PRACTICES USED BY THE ANALYSED PROJECTS DURING THE SOFTWARE DEVELOPMENT 17 10 10 10 10 10 10 10 10 10 10 |
| LIBRA INSTIT 3.1. LIFEC 3.2. LIFEC 3.3. 3.4. DEVE 3.5. 4. AN | RIES AND TOOLS USED IN THE PROJECTS ANALYSED FROM THE EUROPEAN UTIONS |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

| 4.2. | SOF | TWARE DEVELOPMENT LIFECYCLE | 83 |
|------|------|---|----|
| 4.2. | 1. | Software Development Lifecycle Methodologies | 83 |
| 4.2. | 1.1. | Methodologies | 83 |
| 4.2. | 1.2. | Tools | 85 |
| 4.2. | 1.3. | Conclusion | 85 |
| 4.2. | 2. | Security Requirements | 85 |
| 4.2. | 2.1. | Security Requirements Definition | 85 |
| 4.2. | 2.2. | Conclusion | 86 |
| 4.2. | 3. | Testing and Validation | 87 |
| 4.2. | 3.1. | Automatic Testing | 87 |
| 4.2. | 3.2. | Security testing | |
| 4.2. | 3.3. | Validation Testing | |
| 4.2. | 3.4. | Tools and methods | |
| 4.2. | 3.5. | Conclusion | |
| 4.2. | 4. | Release Management | 90 |
| 4.2. | 4.1. | Deployment | 90 |
| 4.2. | 4.2. | Continuous testing and validation | |
| 4.2. | 4.3. | DevOps | 93 |
| 4.2. | 4.4. | Release Planning | 93 |
| 4.2. | 4.5. | Channels and tools used | 94 |
| 4.2. | 4.6. | Conclusion | 94 |
| 4.2. | 5. | Inspection and code review | |
| 4.2. | 5.1. | Code review | |
| 4.2. | 5.1. | 1. Tools | |
| 4.2. | 5.2. | Projects reviewed by security experts | 95 |
| 4.2. | 5.3. | Phase where the project is reviewed by security experts | |
| 4.2. | 5.4. | Conclusion | 97 |
| 4.2. | 6. | Application authentication and authorisation | 97 |
| 4.2. | 6.1. | Authentication | 97 |
| 4.2. | 6.2. | Authorisation | |
| 4.3. | Prc | DJECT MAINTENANCE | |
| 4.3. | 1.1. | Incident management support and categorisation | |
| 4.3. | 1.2. | Incident resolution | |
| 4.3. | 1.3. | Handling of major incidents | |
| 4.3. | 1.4. | User notification | |
| 4.3. | 1.5. | Channels and tools used | |
| 4.3. | 1.5. | 1. Time to resolve a bug or vulnerability | |
| 4.3. | 1.6. | Conclusion | |
| 4.3. | 2. | Problem Management | |

Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 4 of 110

| 4.3 | 8.2.1. | Identification of security updates or bugs | 105 |
|--------|--------|--|-----|
| 4.3 | 8.2.2. | Problem resolution plan | 106 |
| 4.3 | 8.2.3. | Tools and resources used | 106 |
| 4.3 | 8.2.4. | Conclusion | 106 |
| 4.4. | How E | EUROPEAN INSTITUTIONS CONTRIBUTE TO FOSS COMMUNITIES | 107 |
| 4.5. | Relev | /ANT OPINIONS AND ADVICES FROM INTERVIEWEES | 108 |
| 5. BIE | BLIOGF | RAPHIC REFERENCES | 109 |
| 6. AN | NEXES | 5 | 110 |
| 6.1. | QUES | TIONNAIRE FOR THE INTERVIEW | 110 |
| 6.2. | Execu | JTIVE SUMMARY | 110 |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

List of tables

| Table 4-1 Analysis of identified software development methodologies used in the European Institutions – Project management methodology |
|---|
| Table 4-2 Analysis of identified software development methodologies used in the European Institutions – Software development methodologies (Detail) |
| Table 4-3 Analysis of identified software development methodologies used in the European Institutions – Software development methodologies |
| Table 4-4 Analysis of identified software development methodologies used in the European Institutions – Software development methodologies (cycle) |
| Table 4-5 Analysis of identified software development methodologies used in the European Institutions – Delivery pipeline |
| Table 4-6 Analysis of identified software development methodologies used in the European Institutions – Continuous delivery tools |
| Table 4-7 Analysis of identified software development methodologies used in the European Institutions – Code review teams |
| Table 4-8 Analysis of identified software development methodologies used in the European Institutions – Security experts review |
| Table 4-9 Analysis of identified software development methodologies used in the European Institutions – Security review phase |
| Table 4-10 Analysis of identified software development methodologies used in the European Institutions – Authentication methods |
| Table 4-11 Analysis of identified software development methodologies used in the European Institutions – Authorisation |
| Table 4-12 Analysis of identified software development methodologies used in the European Institutions – Bug and vulnerability reporting |
| Table 4-13 Analysis of identified software development methodologies used in the European Institutions – Major incidents resolution 101 |
| Table 4-14 Analysis of identified software development methodologies used in the European Institutions – Channels and tools for communication 103 |
| Table 4-15 Analysis of identified software development methodologies used in the European Institutions – Identification of security updates or bugs 105 |

| Table 4-16 Analysis of identified software development methodologies used in the European Institutions – |
|--|
| Problem resolution plan |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

List of Figures

| Figure 1. Methodological approach to build the analysis- Information sources14 |
|---|
| Figure 2. Analysis of identified software development methodologies used in the European Institutions – Usage of PM for EP methodology in the European Parliament |
| Figure 3. Analysis of identified software development methodologies used in the European Institutions – Usage of PM2 methodology in the European Commission |
| Figure 4. Analysis of identified software development methodologies used in the European Institutions – Methodologies |
| Figure 5. Analysis of identified software development methodologies used in the European Institutions – Security requirements |
| Figure 6. Analysis of identified software development methodologies used in the European Institutions – Automatic tests |
| Figure 7. Analysis of identified software development methodologies used in the European Institutions – Automatic security tests |
| Figure 8. Analysis of identified software development methodologies used in the European Institutions – Validating testing flow |
| Figure 9. Analysis of identified software development methodologies used in the European Institutions – Delivery pipeline |
| Figure 10. Analysis of identified software development methodologies used in the European Institutions – Continuous deployment |
| Figure 11. Analysis of identified software development methodologies used in the European Institutions – Roadmap definition |
| Figure 12. Analysis of identified software development methodologies used in the European Institutions – Incident supporting levels |
| Figure 13. Analysis of identified software development methodologies used in the European Institutions – 1 st level support flow |
| Figure 14. Analysis of identified software development methodologies used in the European Institutions – 2 nd level support flow |
| Figure 15. Analysis of identified software development methodologies used in the European Institutions – 3 rd level support flow |
| Figure 16. Analysis of identified software development methodologies used in the European Institutions – User notification |

| Figure 17. Analysis of identified software development methodologies used in the Europea | n Institutions – |
|--|------------------|
| time to fix bug / vulnerability | 104 |
| Figure 18. Analysis of identified software development methodologies used in the Europea | n Institutions – |
| FOSS communities contribution | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Acronyms and abbreviations

| EI | European Institutions |
|-------|---|
| EP | European Parliament |
| DG | Directorate General |
| FOSS | Free and Open Source Software |
| FOSSA | Free and Open Source Software Auditing |
| os | Operating System |
| SDLC | System Development Life Cycle |
| SEO | Search Engine Optimization |
| WP | Work Package |
| ΑΡΙ | Application Programming Interface |
| ESAPI | Enterprise Security Application Programming Interface |
| | |
| | |
| | |
| | |
| | |
| | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

1. Introduction

1.1. Objective of this Document and Intended Audience

This document represents the deliverable 3 included within TASK-01: Analysis of software development methodologies used in the European Institutions.

The objective of this document is to analyse the software development methodologies, tools and best practices used in the European Institutions' projects that were selected and prioritised in Deliverable 1.

This document is targeted at the DIGIT areas interested in the study of the software development methodologies, related practices and tools used in the European Institutions (European Commission and European Parliament).

1.2. Scope

The analysis covers the European Institutions' projects selected and prioritised during the development of Deliverable 1 and whose project sponsors were interviewed

Throughout the document, the term "European Institutions" refers to the 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 followed to conduct the analysis of the different methodologies, tools and best practices used in the European Institutions' FOSS projects selected, according to the scope.
- Section 3: Software development methodologies, best practices and tools used in the European Institutions.
- Section 4: Analysis of the identified software development methodologies used in the European Institutions.
- Section 5: **Bibliographical** references.
- Section 6: Annexes.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

1.4. Key success factors

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

- To have a complete stock of methodologies used both in the European Institutions and in the FOSS communities that were selected for this project
- The Best practices will include a variety of typologies: technical, organisational and about the governance and quality of open source software (e.g.: synchronisation with FOSS; guidelines for secure software development; secure integration and interoperability of different components; sustainable ways of FOSS governance and professional services).

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

2. Methodological Approach to Building the Analysis

The goal of this document is to analyse all the information gathered during the interviews and the documentation that is relevant for the purpose of the study. This analysis will provide valuable information from the perspective of the European Institutions' projects identified with regard to:

- Software development methodologies in use
- Best practices in use
- Tools in use
- Release management
- Incident management
- Security aspects related to software development
- Additional necessities identified by stakeholders and their points of view regarding how European Institutions can contribute to ensure that the widely used critical software can be trusted.

2.1. Selection of Projects, Engagement with the European Institutions and Information Gathering

Deliverable 1 provided a list of 15 projects to be analysed. Out of the 15 projects, 11 were from the European Commission and the remaining 4 from the European Parliament.

For this step, the following activities were conducted:

- Deliverable 1 provided a list of 15 projects to be analysed.
- To engage with the project owners, the Project Officer from DIGIT sent an executive summary explaining the importance of the FOSSA project, and requesting their availability for an interview to gather information on their particular project.
- The Project Officer from DIGIT developed an interview planning).
- 14 out of 15 projects were covered during the interview rounds.

The information gathering of the 14 projects was conducted in 12 interviews

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

2.2. Information Classification and Filtering Process

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

Figure 1. Methodological approach to build the analysis- Information sources



Interview Results: During the interviews with the European Institutions' project owners, we used a questionnaire to obtain the relevant information for the study. Since the interviews were conducted as an open discussion, 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 methodology, practice or tool under analysis. Only common criteria were taken into account and the analysis does not include what is particular to a project, unless it is relevant for the study. After filtering and classifying the data, each methodology, practice or tool was compared with other projects showing the percentage of usage within the projects analysed.

• **Documentation Analysis**: In order to complete the information related to the methodologies, best practices and tools identified, public documentation was analysed in order to fulfil the aspects mentioned above.

2.3. Analysis of the Information

Sections 3 and 4 of this document are structured following two main purposes:

- Software development methodologies, best practices and tools used in the European Institutions: For each of the methodologies, best practices and tools gathered from the interviews, a form is developed in order to complete the information about each variable.
- Analysis of identified software development methodologies, best practices and tools used in the European Institutions: This section is structured following four main points to conduct the analysis:
 - Software Development Lifecycle: It contains the analysis of methodologies, practices and tools used within the different phases of the project development.
 - Quality Assurance and Maintenance: It analyses the methodologies, practices and tools used to ensure the sustainability of the projects and their quality.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

- **How European Institutions contributes to FOSS Communities**: It analyses the real contribution of the projects and teams to FOSS communities.
- Relevant opinions and advices from interviewees: It contains interviewees' personal opinions and pieces of advices expressed during the interviews.

The usage of each analysed variable is represented by a numeric value and a percentage. To represent these numbers, we used two different approaches:

• **Tables**: Represent the percentage of usage for the total number of projects analysed. It is important to note that the variables are not mutually exclusive; therefore, a project can use one or more of them.

To calculate this percentage, we used the following formula:

%usage = nCoincidences * 100 / nProjectsAnalysed

Pie charts: These charts allow a quick reading of the results since the percentages of usage are represented graphically. The variables analysed using this approach are exclusive; therefore, a project can only use one of them.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

3.Software Development Methodologies, Best Practices, Frameworks, Libraries and Tools Used in the Projects Analysed from the European Institutions

3.1. Methodologies Used by the Analysed Projects During the Software Development Lifecycle

| M1. Met | nodology | Name: | Р | M ² | | | | | |
|-----------------------|--|---|--|-----------------|---------|--|--------|--------------|---|
| Use | | es Benefits | | | | | | | |
| Project mana | gement. | methodolo Project Ma solutions a European through the | A project Governan Structure tailored to European Commission PM2 process guided Artefact Templates. A set of effective Missets. | | | red to the nmission guidelines lates. | | | |
| | | SDLC P | hase | e Where It Is U | sed | | | | |
| Analysis | C | Design | D | evelopment | Testing | | ng | g Deployment | |
| Х | | Х | | Х | Х | | | Х | |
| Roles (i.e. PN | /I, develop | er, etc.) | Project manager | | | | | | |
| | Related Methodologies, Best Practices and Tools | | | BOK | | | | | |
| Related Tech | nologies | | | | | | | | |
| | Proje | ct 1 | Х | Project 5 | | х | Projec | xt 9 | |
| Project Using Project | | ct 2 | х | Project 6 | | | Projec | xt 10 | |
| This Methodolog | y Proje | ct 3 | Х | Project 7 | | Х | Projec | xt 11 | Х |
| | ct 4 | | Project 8 | | Х | Projec | xt 12 | х | |

| M2. | Method | ology | Name: | Р | M for EP | | | | | |
|------------------------|--------------------------|--------|--|-----------------|-----------------|---------------------|-------|------------|------------|--|
| Use | | | Objective | s Benefits | | | | | | |
| n F S E tł | | | The objective of this methodology is to enable Project Managers to deliver solutions and benefits to the European Parliament through an effective project management methodology | | | Parliament projects | | | | |
| SDLC PI | | | | | e Where It Is U | lsed | | | | |
| Anal | Analysis Design | | | D | evelopment | Testing | | | Deployment | |
| > | < | | Х | | х | Х | | | Х | |
| Roles (i | .e. PM, d | evelop | er, etc.) | Project manager | | | | | | |
| Related Practice | Metho es and To | | es, Best | ΡM | BOK | | | | | |
| Related | Techno | logies | | | | | | | | |
| Project 1 | | | | Project 5 | | х | Proje | ct 9 | | |
| Project Using Project | | ct 2 | | Project 6 | X Pro | | Proje | ject 10 | | |
| | his odology Project 3 | | ct 3 | | Project 7 | Pro | | Proje | ect 11 | |
| | | Proje | ct 4 | Х | Project 8 | F | | Project 12 | | |

| МЗ. | Method | ology | Name: | Scrum | | | | | | | |
|-----------------------|--|---------|-----------|--|-----------------|---------|--|------------|------|---|--|
| Use Objectives | | | | | s Bene | | | efits | | | |
| develo | | | | e the software oment using an e and incremental ethod | | | This methodology maximizes the team's ability to deliver quickly, to respond to emerging requirements and to adapt to evolving technologie and changes in market conditions. | | | | |
| SDLC P | | | | | e Where It Is U | lsed | | | | | |
| Anal | Analysis Design | | esign | Development | | Testing | | Deployment | | | |
| × | (| | х | Х | | Х | | х | | | |
| Roles (i | .e. PM, d | evelope | er, etc.) | Scrum Master, Product Owner, Development Team Member | | | | | | | |
| | Related Methodologies, Best Practices and Tools | | | | le | | | | | | |
| | Project 1 | | Х | Project 5 | | | Project 9 | | x | | |
| Project Using Project | | 2 | Х | Project 6 | | | Project 10 | | | | |
| Methoo | | Project | 3 | | Project 7 | | х | Project | : 11 | | |
| | | Project | 4 | Х | Project 8 | | | Project | : 12 | Х | |

| M4. | Method | lology | Name: | Α | gile@EC | | | | | |
|----------------------|-----------------------|-----------------|---|--|-----------------|--|--------|--------|------------|---|
| Use Objectiv | | | | | s Benefits | | | | | |
| Software development | | | Manage the software development using an iterative and incremental agile method. | | | An agile method tailored to European Commission projects | | | | |
| | | | SDLC PI | nase | e Where It Is U | lsed | | | | |
| Ana | Analysis Design | | | D | evelopment | ٦ | Festir | ng | Deployment | |
| > | X | | х | | Х | Х | | | | |
| Roles (i | i.e. PM, d | evelope | er, etc.) | Business analyst, system architect, test architect, project manager, tester, developer | | | | | | |
| Related Practice | Metho es and To | odologi ools | es, Best | Agi | le | | | | | |
| Related | l Techno | logies | | | | | | | | |
| | Project 1 | | | | Project 5 | | Projec | | xt 9 | |
| | Project Using Project | | xt 2 | Х | Project 6 | | Projec | | t 10 | Х |
| | nis dology | Project 3 | | _ | Project 7 | | Projec | | ct 11 | |
| | | Projec | xt 4 | | Project 8 | | | Projec | t 12 | |

| M5. | Method | lology l | Name: | к | anban | | | | | |
|---------------------|-------------------------------|-----------------|--|--|--|----------------|----------------|--|--------------------|-----|
| Use | | | Objectives | S | | Bene | fits | | | |
| Softwar | e develop | oment | Drive the s developme through we limited pha ensures th and the qu artefact go phase. | ent p ell-de ases e co ality | orocess efined and , which ompleteness when an | class devel | ificat Iopm | ne quali ion for t ent of a bottlen | he Irtefacts ai | nd |
| | | | SDLC PI | hase | e Where It Is U | lsed | | | | |
| Anal | ysis | D | esign | D | evelopment | ٦ | estir | ng | Deploym | ent |
| > | K | | х | | Х | | Х | | Х | |
| Roles (i | .e. PM, d | evelope | er, etc.) | No | existing roles. | (The h | nelp o | of an ag | ile coach) | |
| Related Practice | Metho es and To | odologi ools | es, Best | Agi | le | | | | | |
| Related | Techno | logies | | | | | | | | |
| | | Projec | xt 1 | Х | Project 5 | | | Projec | xt 9 | |
| | Project Using Project 2 | | | | Project 6 | | | Projec | xt 10 | |
| | This Methodology Project 3 | | | | Project 7 | | Х | Projec | xt 11 | |
| | | Projec | xt 4 | | Project 8 | | | Projec | xt 12 | |

| M6. Met | nodology | Name: | W | /aterfall | | | | |
|-----------------------------|---------------------------------|---|--|---|---|---|--|------------------------------|
| Use | | Objectives | 5 | | Benefits | \$ | | |
| Software dev | elopment. | phase cons tasks with o objectives. | evel livide nase sists diffe All 1 and | opment ed into s and each of series of rent these phases they have to | rigidity of phase ha and a re are proc | of the m as specifi eview pro cessed a | age due to nodel – e ic deliveral ocess. Pha nd comple d they car | each bles ises eted |
| | | SDLC Pr | nase | e Where It Is U | lsed | | | |
| Analysis | C | Design | D | evelopment | Tes | ting | Deploym | ient |
| Х | | Х | | Х | X | (| Х | |
| Roles (i.e. PN | /I, develop | er, etc.) | | ject manager veloper, tester, | | | vst, archit | ect, |
| Related Me Practices and | thodologi d Tools | es, Best | Wa | terfall | | | | |
| Related Tech | nologies | | | | | | | |
| | Proje | ct 1 | | Project 5 | | Projec | ct 9 | |
| | Project Using Project 2 This | | | Project 6 | | Projec | ct 10 | |
| Methodolog | y Proje | Project 3 | | Project 7 | | Projec | ct 11 | Х |
| | Proje | ct 4 | | Project 8 | | Projec | ct 12 | |

| M7. | Method | lology l | Name: | R | UP@EC | | | | | |
|---------------------|---------------------------------|----------------|--|--|---|-----------|-----------------------|---------------------------------|------------|------|
| Use | | | Objectives | s | | Benefi | ts | | | |
| Softwar | e develoj | oment. | Unified Pro by Rationa Corporatio software d methodolo | oces Il So In [1] evel gy ta Corr erativ | l. It is a opment ailored to the nmission that /e and | list of a | artef enta atio | acts to ation ar n of the | ; | |
| | | | SDLC PI | hase | e Where It Is U | lsed | | | | |
| Anal | lysis | D | esign | D | evelopment | Те | stin | g | Deploym | ent |
| > | K | | х | | Х | | Х | | Х | |
| Roles (i | .e. PM, d | evelope | er, etc.) | | oject manager veloper, tester, | | | | st, archit | ect, |
| Related Practice | Metho es and To | dologi ools | es, Best | RU | Ρ | | | | | |
| Related | Techno | logies | | | | | | | | |
| | | Projec | rt 1 | | Project 5 | | | Projec | xt 9 | |
| | Project Using Project 2 This | | | Х | Project 6 | | | Projec | xt 10 | |
| | dology | Projec | xt 3 | | Project 7 | | | Projec | x 11 | |
| | | Projec | xt 4 | | Project 8 | | | Projec | xt 12 | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

3.2. Best Practices Used by the Analysed Projects During the Software Development Lifecycle

| BP1. Be | st Pra | actice | e Name | e : | | Se | curit | y iı | n De | esigi | n Pl | hase | |
|---|-----------------------------------|--------------------------|---------------|---------------------------|-------------------------|---|---------------------|------|--------------------|---------------|---------------------|--|------|
| Use | | | | Ob | jecti | ves | | | Be | nefit | ts | | ľ |
| Security ex involved in design to p design flav awareness security ris | appl preve vs, or s of p | icatio nt pos prov | ssible ide | sec Det risk the | tect p s, ac appl | oossil flaws oossil ccordi licatic ture. | s. ble ing to | , | des App of p | sign oly c | of p our ible | ations by bossible risks htermeasure security ies. | |
| | | | SDLO | C Ph | ase | Whe | e It I | s U | sed | | | | |
| Analysis | Des | ent | Tes | ting | D | eplo | yme | ent | Maintenan | ice | | | |
| X | X X N/A | | | | | | | | Ν | /A | | N/A | |
| Roles (i.e. | Anal | yst, d | evelop | er, te | ester |) | (Se Arcl | | | Ar | naly | st, (Secur | ity) |
| Related Practices | | | dologie | es, | | Best | | | | | | | |
| Related To | echn | ologi | es (i.e. | Jav | va) | | | | | | | | |
| | Project 1 | | | | | Proj | ect 5 | | | Х | Pr | oject 9 | |
| _ | Projects Using This | | | | | Proj | ect 6 | | | | Pr | oject 10 | |
| Best | | Proje | ect 3 | | х | Proj | ect 7 | | | Х | Pr | oject 11 | |
| | | Proje | ect 4 | | х | Proj | ect 8 | | | Х | Pr | oject 12 | |

| BP2. Bes | st Pr | actice | e Name | e: | | Ex | plici | t So | ecu | rity I | Req | Juire | ments | |
|---|----------------------------------|---------------|-------------|------------------|-------|------|--------------|------|-------------------|-------------------------|-----------------------|------------------------------------|---------|-------|
| Use | | | | Ob | jecti | ves | | | Be | nefi | ts | | | |
| Add explicit requirement requirement phase | nts in nts ga | the atheri | U | me enh app | | ion | | | aga app Imp | ainst olica orove | : po: tion e ap | nterm ssible risks oplica | S. | es |
| Enforce the security function the application of the security function | nctio | | | | | | | | seo | curity | / | | | |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | lsed | | | | | |
| Analysis | Analysis Design Developmer | | | | | | | | eplc | yme | ent | Ma | intenar | nce |
| X | 2 | x | ١ | J/A | | N | /A | | N | /A | | | N/A | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | ·) | (seo arch | | | Ar | naly | st, | (secu | rity) |
| Related Practices | | | dologi S | es, | ł | Best | | | | | | | | |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | | | | | | | | |
| | | Proje | ect 1 | | | Proj | ect 5 | | | Х | Pr | oject | t 9 | |
| | Projects Using This Project 2 | | | | | Proj | ect 6 | | | Х | Pr | oject | t 10 | х |
| Best | | Proje | ect 3 | | х | Proj | ect 7 | | | Х | Pr | oject | 11 | |
| | | Proje | ect 4 | | х | Proj | ect 8 | | | Х | Pr | oject | 12 | |

| BP3. Bes | t Practi | ce Nam | e: | | Ro | ole-ba | ase | ed A | utho | oris | ation | | |
|---|---|----------------|---------|-----------------|---------------------------|--------|-----|--------------------------|---------------------------------|----------------------------|---|-------------|--|
| Use | | | Obj | ecti | ves | | | Ве | nefi | ts | | | |
| Authorisation roles that p easy way to privileges to according to profiles. Th centralized of user privi | rovides o grant o users o differe is allows manage | an ent user | user | r priv ordir | zatior vilege ng to | s | | res sec Us to t | sourc cure ers c the r | es i way only eso | application is granted y. have acce urces defir profile. | in a ess | |
| | SDLC Phase Where It Is Used | | | | | | | | | | | | |
| Analysis | Design | Deve | lopme | ent | Tes | ting | D | eplo | byme | ent | Maintena | ance | |
| x | Х | | Х | | N/ | Ά | | Ν | I/A | | N/A | | |
| Roles (i.e. | Analyst | develop | oer, te | ster |) | Ana | lys | t, ar | chite | ect, | developer | | |
| Related Practices a | | odologi ols | es, | 8 | Best | | | | | | | | |
| Related Te | chnolo | gies (i.e | . Java | a) | | | | | | | | | |
| | Pr | oject 1 | | Х | Proje | ect 5 | | | | Pr | oject 9 | x | |
| Projects Using Thi | | oject 2 | | Х | Proje | ect 6 | | | Х | Pr | oject 10 | х | |
| Best | | | | | Proje | ect 7 | | | Х | Pr | oject 11 | x | |
| | Project 4 | | | Х | Proje | ect 8 | | | Х | Pr | oject 12 | | |

| BP4. Bes | st Pra | ctice | e Name |): | | Sta | anda | rd | Aut | hent | ica | tion Mod | ule | |
|---|---|---------------------|----------|-------------------|--------|-----------------------------|-----------------------|------|------------------|-------------------------|---------------------|---|---------------|-----|
| Use | | | | Ob | jecti | ves | | | Ве | nefi | ts | | | |
| Use a robu authenticat is based or authenticat instead of custom one | tion me n comi tion pr creatin | odul mon otoc | e that | aut Avo aut | oid ty | ication pical ication | | | me and use | echar d onl ers a | nisn y a re a | ntication n is secure uthorized able to acc tion. | · | |
| | | | SDLC | C Ph | ase | Wher | e It I | s U | sec | | | | | |
| Analysis | Desi | gn | opm | ent | Tes | ting | D | eplo | byme | ent | Mainten | ance | | |
| N/A | Х | Х | | N/ | Ά | | Ν | I/A | | N/A | L. | | | |
| Roles (i.e. | Analy | ′st, d | evelop | er, te | ester | ·) | Ana | lys | t, de | evelo | per | , architec | t | |
| Related Practices | | | dologie | es, | E | Best | Alfre secu Mine | urit | у, | | | rity, S CAS, | pring Site | · . |
| Related Te | echno | logi | es (i.e. | Jav | 'a) | | | | | | | | | |
| | ł | Proje | ect 1 | | x | Proje | ect 5 | | | Х | Pr | oject 9 | X | |
| | Projects Using This Project 2 | | | | | Proje | ect 6 | | | | Pr | oject 10 | X | |
| Best Practice | | Proje | ect 3 | | Х | Proje | ect 7 | | | Х | Pr | oject 11 | X | |
| | | Proje | ect 4 | | х | Proje | ect 8 | | | Х | Pr | oject 12 | X | |

| BP5. Be | st Pr | actice | e Name | 9: | | Sta | anda | rd / | Aut | horis | sati | on Module | |
|--|----------------------------------|---------|--------------|--------------------|-------|-------|--------|------|------------------------------|-----------------------------------|--------------------------|--|------|
| Use | | | | Ob | jecti | ves | | | Be | nefit | s | | |
| Use a robu Authorisati instead of custom on | ion m creat | odule | | con Avc priv | trol. | | SS | | me en apj en use | echan ough plicat abling | isn to ion g ao | risation n is secure protect resources, ccess only f correspond | or |
| | | | SDLO | C Ph | ase | Wher | e It I | s Us | sec | | | | |
| Analysis | Analysis Design Developmen | | | | | | | | eplo | ymei | nt | Maintenan | ice |
| N/A | | | | | | | | | Ν | I/A | | N/A | |
| Roles (i.e. | Ana | lyst, c | levelop | er, te | ester | ·) | Ana | lyst | , de | evelo | per | , architect | |
| Related Practices | | | dologie S | es, | l | Best | Alfro | esco | o se | ecurit | y, S | Spring secu | rity |
| Related T | echn | ologi | es (i.e. | Jav | a) | | | | | | | | |
| | Project 1 | | | | | Proje | ect 5 | | | | Pr | oject 9 | X |
| | Projects Using This Project 2 | | | | Х | Proje | ect 6 | | | | Pr | oject 10 | х |
| Best | | Proje | ect 3 | | | Proje | ect 7 | | | | Pr | oject 11 | |
| | | Proje | ect 4 | | | Proje | ect 8 | | | | Pr | oject 12 | х |

| BP6. Be | st Practi | ce Nam | e: | St | anda | rd (| Cry | ptog | Irap | hic Modul | e |
|---|---------------------|---------------|------------------------------|------|--------|------|-----------|-------------|--------------|---|-----|
| Use | | | Object | ives | | | Be | nefi | ts | | |
| Use a robu cryptograp instead of custom on | hic mod creating | ule, | Robust algorith implem | m | • | | coi Te | mmu sted | nica libr | plication ations. ary for critic managem | |
| | | SDL | C Phase | Whe | e It I | s U | sec | | | | |
| Analysis | Design | Deve | lopment | Tes | ting | De | eplc | yme | nt | Maintenar | nce |
| N/A | Х | Х | N | /A | | Ν | I/A | | N/A | | |
| Roles (i.e. | Analyst | develop | oer, teste | r) | Ana | lys | t, D | evelo | ре | r, Architect | |
| Related Practices | | odologi Is | es, | Best | Оре | enS | SL | | | | |
| Related Te | echnolo | gies (i.e | . Java) | | | | | | | | |
| | Pr | oject 1 | | Proi | ect 5 | | | | Pr | oject 9 | |
| Project | <u>ه</u> | | | - | | | | v | | • | |
| Using Th Best | | oject 2 | | Proj | ect 6 | | | Х | Pr | oject 10 | |
| Practice | e Pr | oject 3 | | Proj | ect 7 | | | | Pr | oject 11 | Х |
| | Pr | oject 4 | | Proj | ect 8 | | | | Pr | oject 12 | |

| BP7. Bes | st Pr | actice | e Name | e: | | We | ell-te | ste | d B | ase | Тес | hnology | |
|--|--|------------------------------|---------------------|---|---|---------------|-------------------------------|------|--------------------------------|--|---------------------------------------|---|----------|
| Use | | | | Ob | jecti | ves | | | Be | nefi | ts | | |
| Use a robu underlying application that has be many prod environme widely use | tech deve en te uctio nts, a | nolog elopm ested n | y for ient in | vuli Act of t tec pla sec fun | neral ive c he b hnolo ce to curity ction | ogy gi new | isks. pme ives s and | nt | pre reg une Re pos | eviou gardi derly duci ssibl | is ex ng s ring ng t e se | ntage of the xperiences security of th technology. he number ecurity flaws echnology. | ne of |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | lsed | 1 | | | |
| Analysis | Des | sign | opm | ent | Tes | ting | D | eplo | byme | ent | Maintenar | nce | |
| N/A |) | X | | Х | | N/ | /A | | Ν | I/A | | N/A | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | .) | Ana | lys | t, de | evelo | per | , architect | |
| Related Practices | | | dologie S | es, | l | Best | | | | | | | |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | Java | a, | PHF | D | | | |
| | | Proje | ect 1 | | х | Proje | ect 5 | | | Х | Pr | oject 9 | x |
| | Projects Using This Project 2 | | | | Х | Proje | ect 6 | | | Х | Pr | oject 10 | Х |
| Best | | Proje | ect 3 | | Х | Proje | ect 7 | | | Х | Pr | oject 11 | Х |
| | | Proje | ect 4 | | х | Proje | ect 8 | | | Х | Pr | oject 12 | х |

| BP8. Bes | st Pr | actic | e Name | e : | | | ecurit kterna | | | Awa rces | ren | ess | Fr | om |
|---|---|---|---------------------------|---------------------------------|--|--|------------------|-----|------------|------------------|---------------|--------------------------------------|-------------------|-----|
| Use | | | | Ob | jecti | ves | | | Be | nefit | s | | | |
| Many softw FOSS com provide info security iss software th Also public repositorie information flaws. The should che information possible se the applica | imun orma sues hey d vulr s sha s sha n of s proje ck th n to k ecurit | ities tion a of the evelo erabi are ecurit ect tea now o ty risk | ibout p. lity am | sec sof the (co app | tware tware app mpo plicat | [,] issu e use licati nent | 8, | | seo the | curity e requ | / ris uire | e of po ks wh d prev be tak | ich all /entiv | low |
| | | | C Ph | ase | Whe | re It I | s U | sec | | | | | | |
| Analysis | De | sign | Devel | opm | ent | Te | sting | D | eplo | byme | nt | Mair | ntenar | nce |
| N/A | 2 | x | | Х | | N | I/A | | Ν | I/A | | | Х | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | estei | ⁻) | Proj adm | | | Ma tor | nag | jer, | syst | tem |
| Related Practices | | | dologie S | es, | | Best | | | | | | | | |
| Related Te | Related Technologies (i. | | | | | | | | | | | | | |
| | | Proi | ect 1 | | | Pro | ect 5 | | | | Pr | oject s | 2 | X |
| | Projects Project 2 | | | | Х | | ject 6 | | | | | oject · | | x |
| Best | Best Practice | | | | × | | ject 0 | | | Х | | oject ' | | |
| Practice | | | | | | | | | | | | | | v |
| | | Proj | ect 4 | | Х | Pro | ect 8 | | | Х | P٢ | oject ' | 12 | Х |

| -BP9. Best Pi | actice N | ame: C | ontir | nuo | us Testing | | | | | |
|---------------------------------|--------------------|--|-------|-------|--------------------------|--------------------------|-------|------|---|-----|
| Use | 0 | bjectives | | | Benefits | | | | | |
| Release management | at as so | xecute utomated to part of th oftware de peline. | e | , | problems f stage of S | from p DLC. e time | orogr | essi | can prevent ng to the ne ort needed t | |
| | | SDLC P | hase | W | here It Is U | sed | | | | |
| Analysis | Design | Develo | opme | nt | Testing | Dep | loym | ent | Maintenan | ice |
| N/A | N/A | | Х | | Х | | N/A | | N/A | |
| Roles (i.e. An tester) | nalyst, d | developer | , Te | ste | r, operation | S | | | | |
| Related Meth Practices and T | odologie ools | s, Best | Co | ontii | nuous deliv | ery, c | ontin | uou | s deploymei | nt |
| Related Techno | logies (i. | e. Java) | | | | | | | | |
| | Project | 1 | x | Pro | oject 5 | | | Pro | oject 9 | x |
| Projects | Projects Project 2 | | | | oject 6 | | х | Pro | oject 10 | |
| Using This Best Practice | Project | 3 | Х | Pro | oject 7 | | Х | Pro | oject 11 | Х |
| | Project | 4 | Х | Pro | oject 8 | | | Pro | oject 12 | x |

| BP10. Bes | Co | Code Review | | | | | | | | | |
|--|-------------|-------------|--|-----------|------------|------|--|------------|-------------|---|--|
| Use | Objectives | | | | Benefits | | | | | | |
| Developed code is analysed in order to review possible security issues, bugs or standard non- compliant code, during development and testing phases. | | | Find possible application vulnerabilities, bugs or standard non-compliant code | | | | Early bug and vulnerability finding. Improve the code quality. | | | | |
| SDLC Phase Where It | | | | | | | sed | | | | |
| Analysis | Design | Development | | | sting Depl | | eployment | | Maintenance | | |
| N/A | N/A | N/A X | | | x | | N/A | | N/A | | |
| Roles (i.e. A | Analyst, de | velope | r, teste | r) | Test | ter, | develo | oper | | | |
| Related Me and Tools | thodologi | ies, Be | est Pra | ctices | | | | | | | |
| Related Technologies (i.e. Java) | | | | | | | | | | | |
| | Projec | Project 1 | | | Project 5 | | | Pro | ject 9 | х | |
| Projects Using This | | roject 2 | | Proje | Project 6 | | Х | Project 10 | | х | |
| Best Practice | | Project 3 | | Proje | Project 7 | | | Pro | ject 11 | х | |
| | Projec | Project 4 | | Project 8 | | | | Pro | х | | |

| BP11. Best Practice Name: DevOps | | | | | | | | | | | | |
|---|---|---|---------------------------------------|---|-----------|------------|-------------|--|--|--|--|--|
| Use | Obje | | Benefits | | | | | | | | | |
| Release management | oper are i the b prac the a betw tean | t se two gate n the and | resolution Mitigate th manageme | Technical benefits such as faster resolution of problems Mitigate the risks related to the releas management phase | | | | | | | | |
| SDLC Phase Where It Is Used | | | | | | | | | | | | |
| Analysis | Design | Develo | pment | t Testing | Deploy | ment | Maintenance | | | | | |
| N/A | N/A | < | Х | X X N/A | | | | | | | | |
| Roles (i.e. A tester) | nalyst, de | eveloper, | IT te | eam | | | | | | | | |
| Related Meth Practices and T | odologies, ools | Continuous deployment, continuous delivery, agile | | | | | | | | | | |
| Related Techno | logies (i.e. | Java) | | | | | | | | | | |
| | Project 1 | Pr | roject 5 | | Project 9 | | | | | | | |
| Projects Using This Best Practice | Project 2 | | Pr | roject 6 | | Proje | ect 10 | | | | | |
| | Project 3 | | Pr | roject 7 | Х | Proje | Project 11 | | | | | |
| | Project 4 | | Pr | roject 8 | | Project 12 | | | | | | |

| BP12. Best Pr | actice Nar | Use Test | of ing | Non-produ | uction | Envir | onments | For | | |
|--|---|------------------------|-----------|--|---------------|--------|---------------|--------|------|--|
| Use | es | Benefits | | | | | | | | |
| Validation and te | ftware velopm ting sh done o ductio vironm | nould i on the n | not | Development and testing on test environments avoids interfering with the production and their users. It also avoids the risk if wrong or improper actions are performed, or the possibility of new bugs resulting from new functionalities | | | | | | |
| | ed | | | | | | | | | |
| Analysis | Design | Dev | velopn | nent | Testing | Deploy | ment Maintena | | ance | |
| N/A | N/A | Х | | Х | х | | N/A | | | |
| Roles (i.e. A tester) | nalyst, d | evelop | oer, | Teste | r, operations | 5 | | | | |
| Related Methodologies, Best Practices and Tools | | | | | | | | | | |
| Related Techno | logies (i.e. | Java) | | | | | | | | |
| | Project 1 | | | Proj | ect 5 | x | Proje | ect 9 | х | |
| Projects Using This Best Practice | Project 2 | roject 2 | | Proj | ect 6 | x | Proje | ect 10 | Х | |
| | Project 3 | ct 3 | | Proj | ect 7 | x | Proje | ect 11 | Х | |
| | Project 4 | ect 4 | | Proj | ect 8 | Х | Proje | Х | | |

| BP13. Best Practice Name: | | | | | | Initial Assessment Of Release Components | | | | | | | | |
|---|--|------------------|--------|----------|---|---|--|-----|-----------|------------|------|-------------|---|--|
| Use Objectiv | | | | | ives | | | Be | nefi | ts | | | | |
| Validation | Conduct an initial quality assessment of the release components. | | | | This process ensures that the components meet the defined quality criteria to enter the testing phase | | | | | | | | | |
| | SDLC Phase Where It Is Used | | | | | | | | | | | | | |
| Analysis | Desi | sign Development | | | | sting Depl | | | eployment | | | Maintenance | | |
| N/A | N/A | I/A X | | | X | | | N/A | | | N/A | | | |
| Roles (i.e. | Analy | st, dev | velope | r, teste | r) | Developer, tester | | | | | | | | |
| Related Methodologies, B Practices and Tools | | | | | Best | st Validation and testing, testing | | | | | unit | | | |
| Related Technologies (i.e. Java) | | | | | | | | | | | | | | |
| Project 1 | | | t 1 | X | Proj | ect 5 | | | Х | Project 9 | | X | | |
| Projects Using This Best Practice | | Project 2 | | X | Proj | Project 6 | | | Х | Project 10 | | Х | | |
| | | Project 3 | | X | Proj | Project 7 | | | Х | Project 11 | | 11 | Х | |
| | | Project 4 | | X | Project 8 | | | | Х | Project 12 | | | Х | |

| BP14. Best Practice Name: | | | | | Release Testing | | | | | | | | |
|-----------------------------|---|---|------------|------------|---|-----|-----------|----------|-------------|---|--|--|--|
| Use O | | | | Objectives | | | | Benefits | | | | | |
| Validation a | Submitting the release components to intensive tests | | | | This process ensures that only components which meet the quality criteria can be deployed to production | | | | | | | | |
| SDLC Phase Where It Is Used | | | | | | | | | | | | | |
| Analysis | Design | gn Development . | | | ting | D | eploym | ent | Maintenance | | | | |
| N/A | N/A | ١ | N/A | /A X | | | Х | | N/A | | | | |
| Roles (i.e. | Analyst, c | levelop | er, tester | .) | Dev | elo | oper, tes | ster | | | | | |
| Related Practices a | Best | st Validation and testing, un testing, integration testing functional testing, non-functiona testing | | | | | | | | | | | |
| Related Te | chnologi | es (i.e. | Java) | | | | | | | | | | |
| Project 1 X | | | | | roject 5 | | XF | | oject 9 | x | | | |
| Projects Using Th | | Project 2 | | Proje | Project 6 | | Х | Pi | oject 10 | x | | | |
| Best Practice | Proi | Project 3 | | Proje | Project 7 | | Х | Pi | oject 11 | Х | | | |
| | Proj | Project 4 | | Project 8 | | | X Pr | | oject 12 | Х | | | |

| BP15. B | est Prac | tice Nar | ne: | | Us | er A | cce | epta | nce | Tes | sting | |
|----------------------|-----------|---------------|---------------------|-----------------------|--|---------------|------|-------------------|---------------|----------------------|---|--------|
| Use | | | Ob | jecti | ves | | | Be | nefi | ts | | |
| Validation | and testi | ng | rele per or b | ease form ousin | ing th of tes ed by ness Iders | sts ⁄ usei | rs | app fun imp | orov ctior | e th nalit ent | stakeholde e quality a ies ed in the | |
| | | SDL | C Ph | ase | Wher | e It I | s U | lsed | | | | |
| Analysis | Design | lopm | ent | Tes | ting | D | eplo | yme | ent | Maintena | ince | |
| N/A | N/A | N/A | | > | < | | 2 | x | | N/A | | |
| Roles (i.e. | Analyst | develop | oer, te | ester | .) | Dev | elc | oper, | tes | ter | | |
| Related Practices | | odologi Is | es, | l | Best | Vali acc | | tion ance | e tes | an sting | | sting, |
| Related T | echnolo | gies (i.e | . Jav | va) | | | | | | | | |
| | Pr | oject 1 | | х | Proj | ect 5 | | | Х | Pr | oject 9 | x |
| Project Using Th | | oject 2 | | Х | Proj | ect 6 | | | Х | Pr | oject 10 | Х |
| Best Practic | Dr | oject 3 | | Х | Proj | ect 7 | | | Х | Pr | oject 11 | Х |
| | Pr | oject 4 | | Х | Proj | ect 8 | | | Х | Pr | oject 12 | Х |

| BP16. Best P | ractice N | ame: | Auto | omatio | on Test | ing | | | | |
|---------------------------------|-------------------|---|------|---------|-----------|-------|--------------------------------|--------|----------------------------|-----|
| Use | C | Objectives | 5 | | | Ber | nefits | | | |
| Validation and te | ŭ w | Execute au vritten test nanual inte | s wi | thout | ly the | effic | ease ef ciency a ware te | nd cov | ness, verage of | |
| | | SDLC P | has | e Whe | ere It Is | Use | d | | | |
| Analysis | Design | n Deve | lopn | nent | Testi | ng | Deploy | ment | Maintena | nce |
| N/A | N/A | 1 | N/A | | Х | | Х | | N/A | |
| Roles (i.e. A tester) | nalyst, | develope | r, I | Develo | oper, te | ster | | | | |
| Related Meth Practices and T | iodologie ools | es, Bes | t | testing | g, fun | ction | 0. | ting, | ing, integra non-functi | |
| Related Techno | logies (i. | e. Java) | | | | | | | | |
| | Project | 1 | | Proj | ect 5 | | | Proje | ect 9 | |
| Projects | | | | | ect 6 | | X | Proje | ect 10 | Х |
| Using This Best Practice | Project | 3 | Х | Proj | ect 7 | | X | Proje | ect 11 | х |
| | Project | 4 | Х | Proj | ect 8 | | Х | Proje | ect 12 | х |

| BP17. Best P | ractice N | ame: | Secu | rity Tes | ting | | | | |
|---------------------------------|------------------|--|---------|--------------------|----------------------|---------------|---------------|------------|-------|
| Use | | Objecti | ves | | Bene | fits | | | |
| Validation and te | 5 | Check t release known vulnera | is free | e of | Confi Authe | dentiality | Integ Auth | orisation, | |
| | | SDLC | Phas | e Wher | e It Is U | sed | | | |
| Analysis | Design | Dev | elopm | ent Te | esting | Deploy | nent | Maintenan | се |
| N/A | N/A | | N/A | | Х | X | | N/A | |
| Roles (i.e. A tester) | nalyst, o | develop | er, D | evelope | er, analy | vst, tester | | | |
| Related Meth Practices and T | odologie ools | es, Be | | enetrat lack/wh | ion to ite testir | esting, ng | vuln | erability | scan, |
| Related Techno | logies (i. | e. Java) |) | | | | | | |
| | Project | 1 | X | Projec | ct 5 | | Pr | oject 9 | x |
| Projects | | | | | ct 6 | Х | Pr | oject 10 | Х |
| Best Practice | Project | 3 | Х | Proje | ct 7 | X | Pr | oject 11 | |
| | Project | 4 | | Projec | xt 8 | | Pr | oject 12 | Х |

| BP18. B | est P | racti | ce Nan | ne: | | Th | ird-p | art | y To | estin | g | | | |
|--|--|---------|-------------|-------|-------|--------------------------|------------|------|-------------------------|-------------------------|------------------------------|----------|--------------|----|
| Use | | | | Ob | jecti | ves | | | Be | nefit | s | | | |
| Applicatior conducted team of se | by a | n exte | | sec | urity | ssible flaws igura | s and | | vul ap De in f | plicat tect the a | bilit tion con ppli | ies in t | on er and | |
| | SDLC Phase Where It Is Used Analysis Design Development Testing Deployment Maintenance | | | | | | | | | | | | | |
| Analysis | Des | ign | Devel | ent | Tes | ting | D | eplo | yme | nt | Maint | enan | се | |
| N/A | N | /A | ١ | | > | < | | | Х | | | Х | | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | ·) | Sec adm | | | | udit | or, | syst | em |
| Related Practices | | | dologi S | es, | | Best | | | | | | | | |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | | | | | | | | |
| | | Proj | ect 1 | | х | Proj | ect 5 | | | | Pr | oject 9 | | х |
| | Projects Jsing This Project 2 | | | | | Proj | ect 6 | | | | Pr | oject 1 | 0 | х |
| Best | Using This | | | | Х | Proj | ect 7 | | | Х | Pr | oject 1 | 1 | |
| | | Proj | ect 4 | | | Proj | ect 8 | | | Х | Pr | oject 1 | 2 | х |

| BP19. B | est F | Practi | ce Nan | ne: | | Re | leas | e P | lan | ning | | | |
|----------------------|-------|---------|--------------|-------|-------|-----------------------|--------|-----|------|----------------|-------|--------------|------|
| Use | | | | Ob | jecti | ves | | | Ве | nefi | s | | |
| Release m | ianag | jemer | nt | rele | | d sch s and ope | | - | | e rele vanc | | e is plannec | l in |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | sec | | | | |
| Analysis | | | | | | | | De | eplc | yme | nt | Maintenan | ce |
| N/A | N | /A | J/A | | N/ | /Α | | | Х | | N/A | | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | .) | Proj | ect | ma | inage | er, c | operations | |
| Related Practices | | | dologie S | es, | | Best | Rele | eas | e m | iana | gerr | ient | |
| Related To | echn | ologi | es (i.e. | Jav | va) | | | | | | | | |
| | | Duci | 1 4 | | V | Duci | 1 5 | | | X | Du | | |
| Destaut | | Proj | ect 1 | | Х | Proje | ect 5 | | | Х | Pr | oject 9 | |
| Project Using Th | | Proj | ect 2 | | Х | Proje | ect 6 | | | | Pr | oject 10 | Х |
| Best Practic | e | Proj | ect 3 | | Х | Proje | ect 7 | | | Х | Pr | oject 11 | |
| | | Proj | ect 4 | | Х | Proje | ect 8 | | | Х | Pr | oject 12 | Х |

| BP20. B | est P | racti | ce Nan | ne: | | Se | curit | y l | ncio | lent | Ма | nagerr | nent. | |
|---|--|------------------------|-------------|-------------------------|--------------------------------|---|-------------------------|-----|------------------|--------------------------|-----------------------|--|--------------|------|
| Use | Security incident nanagement is part of the security plan, where the nstructions about how to espond to incidents are explained. BLC Phase | | | | | | | | Be | nefi | ts | | | |
| manageme security pla instruction | ent is an, w s abc | part here out ho | the w to | def act sec as | ined ions curity well | early respo in cas incid as a c ey sta | se of ents, conta | ct | me mit imj | echai tigate bacts | hisn e the s in | les an n so as e poss case o cident. | s to ible | tive |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | sec | | | | | |
| Analysis | Des | ent | Tes | ting | D | eplo | byme | ent | Main | tenar | nce | | | |
| N/A | N | | N | /A | | | Х | | | Х | | | | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | .) | Sys [.] mar | | | | | trator, LISO. | pro | ject |
| | | | | es, | I | Best | | | | | | | | |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | | | | | | | | |
| | | Proje | ect 1 | | | Proj | ect 5 | | | Х | Pr | oject 9 |) | |
| | Projects Jsing This | | | | | Proj | ect 6 | | | | Pr | oject 1 | 0 | Х |
| Best Practic | | Proje | ect 3 | | | Proj | ect 7 | | | Х | Pr | oject 1 | 1 | |
| | | Proje | ect 4 | | х | Proj | ect 8 | | | Х | Pr | oject 1 | 2 | |

| BP21. B | est Pra | ctice Na | me: | | Pr | oacti | ve | Probler | n Id | lentificatio | n |
|----------------------|-------------------------------------|----------------|-------------------|---------------|--|---------------|-----|----------|---------------|---|-----|
| Use | | | Ob | jecti | ves | | | Benefi | ts | | |
| Problem m | anager | nent | of t by and | the a iden | e the o pplica tifying nerab nce | tion, bugs | 6 | these is | ssue ng fi | le to addres es in advan ixings or ds. | |
| | | SDL | .C Ph | ase | Wher | e It I | s U | sed | | | |
| Analysis | Desig | n Dev | elopm | nent | Tes | ting | D | eployme | ent | Maintenai | nce |
| N/A | N/A | | N/A | | N | /A | | N/A | | Х | |
| Roles (i.e. | Analys | t, develo | per, t | estei | r) | ІТ Т | ear | m | | | |
| Related Practices | | nodolog ols | ies, | | Best | ITIL | [2] | , Proble | m n | nanagemen | ıt |
| Related Te | echnolo | ogies (i. | e. Jav | /a) | | | | | | | |
| Related Te | echnolo | ogies (i. | e. Jav | /a) | | | | | | | |
| | Р | roject 1 | | Х | Proj | ect 5 | | X | Pr | oject 9 | X |
| | Project 1 Projects Using This | | | | Proj | ect 6 | | х | Pr | oject 10 | Х |
| Best Practice | D | roject 3 | | Х | Proj | ect 7 | | х | Pr | oject 11 | Х |
| | Ρ | roject 4 | | х | Proj | ect 8 | | Х | Pr | oject 12 | х |

| BP22. B | Jse Obj Problem management Rec prio prod faci effe SDLC Pha Analysis Design Developme N/A N/A N/A Roles (i.e. Analyst, developer, te Related Methodologies, Practices and Tools | | | | | | | n sat | C ion | ateç | gori | satio | n | and |
|----------------------|--|--------|----------|---------------------|-----------------|------------------|--------|----------|---------------|-------|------|--------|-------------------------------|-------|
| Use | | | | Ob | jecti | ves | | | Ве | nefi | ts | | | |
| Problem m | ianag | jemer | nt | pric pro faci | blem ilitate | e the n in or | | | of t | he in | ncid | ent to | critica o facil olutior | itate |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | lsed | | | | | |
| Analysis | Des | ent | Test | ting | D | eplo | yme | ent | Mair | ntena | nce | | | |
| N/A | | | | | | | | | N | /A | | | Х | |
| Roles (i.e. | Ana | yst, c | levelop | er, te | ester | ·) | IT T | ea | m | | | | | |
| Related Practices | | | | es, | | Best | ITIL | [2] | <u>[</u> , Pr | oble | m n | nanag | geme | nt |
| Related Te | echn | ologi | es (i.e. | Jav | a) | | | | | | | | | |
| | Project 1 | | | | | Proje | ect 5 | | | Х | Pr | oject | 9 | X |
| | Projects Using This Project 2 | | | | | Proje | ect 6 | | | Х | Pr | oject | 10 | х |
| Best | Jsing This | | | | Х | Proje | ect 7 | | | Х | Pr | oject | 11 | Х |
| | | Proje | ect 4 | | х | Proje | ect 8 | | | Х | Pr | oject | 12 | Х |

| BP23. B | est P | racti | ce Nan | ne: | | Inc | ciden | t N | lana | agen | nen | t Suppor | t |
|----------------------|-------------------------------|--------|--------------|-------------------|-----------------|--|----------------|------|---------------|-------|----------|----------------------------|----|
| Use | | | | Ob | jecti | ves | | | Be | nefi | ts | | |
| Incident m | anage | emen | ıt | ma cha rule | annel es foi | and h the t s, ski r incid ement | lls an lent | | | | | nd efficier f incidents | |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | sec | | | | |
| Analysis | Desi | ign | opm | ent | Test | ting | D | eplc | yme | nt | Maintena | ance | |
| N/A | N/ | A | J/A | | N/ | Ά | | Ν | I/A | | Х | | |
| Roles (i.e. | Anal | yst, c | levelop | er, t | ester | .) | IT T | ea | m | | | | |
| Related Practices | | | dologie S | es, | E | Best | ITIL | [2] | <u>l</u> , In | cider | nt m | anageme | nt |
| Related Te | echno | ologi | es (i.e. | Jav | va) | | | | | | | | |
| | | Proje | ect 1 | | х | Proje | ect 5 | | | Х | Pr | oject 9 | X |
| | Project 1 Project 2 Project 2 | | | | | Proje | ect 6 | | | Х | Pr | oject 10 | X |
| Best | Jsing This | | | | Х | Proje | ect 7 | | | Х | Pr | oject 11 | x |
| | | Proje | ect 4 | | х | Proje | ect 8 | | | Х | Pr | oject 12 | х |

| BP24. B | est P | Practi | ce Nan | ne: | | | ciden itego | | atio | | oggi | ing | | and |
|----------------------|----------------------|---------|-------------|---------------------|-------|-----------------------|----------------|-------|----------------|-------|------|---------|---------------------------------|-----|
| Use | | | | Ob | jecti | ves | | | Be | nefi | ts | | | · |
| Incident m | anag | emer | t | pric inci fac | ident | e the in or the | der to | D | the | inci | den | t to fa | critical acilitat olutior | te |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | lsed | | | | | |
| Analysis | Des | ent | Test | ting | D | eplo | yme | nt | Mai | ntena | nce | | | |
| N/A | | | | | | | | | Ν | /A | | | Х | |
| Roles (i.e. | Anal | lyst, c | levelop | er, t | ester |) | Help | ode | esk, | first | leve | el tec | hnicia | ns |
| Related Practices | | | dologi S | es, | l | Best | ITIL | . [2] | <u>]</u> , Inc | cider | nt m | ianag | jemen | it |
| Related T | echn | ologi | es (i.e. | Jav | va) | | | | | | | | | |
| | | Proj | ect 1 | | х | Proje | ect 5 | | | Х | Pr | oject | 9 | X |
| Project Using Th | | Proj | ect 2 | | Х | Proje | ect 6 | | | Х | Pr | oject | 10 | х |
| Best Practic | Jsing This Project 2 | | | | Х | Proje | ect 7 | | | Х | Pr | oject | 11 | х |
| | | Proj | ect 4 | | Х | Proje | ect 8 | | | Х | Pr | oject | 12 | Х |

| BP25. B | est Practi | ce Nan | ne: | | | | | e Incide Support | | Resolutior | n By |
|----------------------|--|-----------|--|-------------|---------------|-------------|-----|---------------------|----------|------------------------------------|------|
| Use | | | Objec | ctiv | /es | | | Benefi | ts | | ľ |
| Incident m | anagemer | nt | First le issues not re or vul | s w late | hich ed to | are bugs | | • | ona | ers' Il issues are his level | Э |
| | | SDLO | C Phas | e V | Vher | e It I | s U | lsed | | | |
| Analysis | Design | Devel | opmen | t | Test | ting | D | eployme | ent | Maintena | nce |
| N/A | N/A | N/A | | N/ | Ά | | N/A | | Х | | |
| Roles (i.e. | Analyst, o | levelop | er, test | er) |) | Help | ode | esk, first | leve | el technicia | ns |
| Related Practices | | | es, | B | lest | ITIL | [2 |], Incide | nt m | nanagemer | nt |
| Related To | echnologi | ies (i.e. | Java) | | | | | | | | |
| | Proj | ect 1 | X | | Proje | ect 5 | | x | Pr | oject 9 | X |
| Project Using Th | Prol | X | | Proje | ect 6 | | Х | Pr | oject 10 | Х | |
| Best | Project 2 Jsing This Best Project 3 | | | | Proje | ect 7 | | х | Pr | oject 11 | Х |
| | Proj | ect 4 | X | | Proje | ect 8 | | Х | Pr | oject 12 | Х |

| BP26. B | Use Obj Incident management Solv have esca first SDLC Pha SDLC Pha Analysis Design Developme N/A N/A N/A Roles (i.e. Analyst, developer, te Related Methodologies, Practices and Tools Related Technologies (i.e. Java | | | | | | ciden Ippor | | Res | olutio | n | By 2 nd L | evel |
|----------------------|--|---------|----------|------------|----------------|--------------------------------|----------------|-----|--------------------|--------|---------------------|--|------|
| Use | | | | Ob | jecti | ves | | | Ве | nefits | | | ľ |
| Incident m | anag | emer | ıt | hav esc | /e be alate | sues en ed fro el sup | m the |) | tec rela vul | hnical | ise bb ilitie | uration ar sues not ugs or es are sol | |
| | | | SDLO | C Ph | ase | Wher | e It I | s U | sed | | | | |
| Analysis | Des | sign | Devel | opm | ent | Tes | ting | De | eplo | ymen | t | Maintena | nce |
| N/A | N | /A | ١ | N/A | | N | /A | | N | I/A | | Х | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | ·) | Spe | cia | lize | d tech | nic | ians | |
| Related Practices | | | | es, | l | Best | ITIL | [2] | , Ind | cident | ma | anagemer | nt |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | | | | | | | |
| | Project 1 | | | | | Proj | ect 5 | | | | Pro | oject 9 | X |
| | Projects Using This Project 2 | | | | | Proj | ect 6 | | | | Pro | oject 10 | |
| Best Practice | | Proj | ect 3 | | Х | Proj | ect 7 | | | | Pro | oject 11 | Х |
| | | Proj | ect 4 | | х | Proj | ect 8 | | | | Pro | oject 12 | |

| BP27. B | est Pr | acti | ce Nan | ne: | | На | ndlir | ng o | of Majo | or In | cidents | |
|--|--------|--------|--------------|-------------|---------------|----------------------------|---------|--------------|---------|-------|--|----|
| Use | | | | Ob | jecti | ves | | | Bene | its | | |
| Incident m | anage | emen | ıt | issi ser | ues v ious | s critic vhich tions | caus | e | to rec | over | the issue and the service ossible. | 2 |
| | | | SDLC | C Ph | ase | Whe | e It Is | s Us | sed | | | |
| Analysis Design Development Testing Deployment Maintenance | | | | | | | | nce | | | | |
| N/A N/A N/A | | | | | | N | /A | | N/A | | Х | |
| Roles (i.e. | Analy | /st, d | levelop | er, te | ester | ·) | ІТ Т | ean | n | | | |
| Related Practices | | | dologie S | es, | E | Best | ITIL | <u>[2]</u> , | Incide | ent m | nanagemer | it |
| Related Te | echno | ologi | es (i.e. | Jav | a) | | | | | | | |
| | | Proje | ect 1 | | х | Proj | ect 5 | | X | Pi | oject 9 | X |
| Projects Using This Project 2 X | | | | | х | Proj | ect 6 | | X | Pi | oject 10 | Х |
| Best | | | | | х | Proj | ect 7 | | X P | | oject 11 | Х |
| | _ | Proje | ect 4 | | х | Proj | ect 8 | | Х | Pi | Х | |

| BP28. | Best Practice Name: Security Incident No. Objectives Benefits | | | | | | | No | tification | | | | |
|---|--|-------------------------------------|-------------------------------|---------------------------|--|--|------------------|------|----------------------------|------------------------|-----------------------------|--|-----------|
| Use | | | | Ob | jecti | ves | | | Be | nefit | ts | | |
| Security in communica inform ther impact and actions for resolution change) | ated n abe l sug cont | to use out th gest s ainme | ers to e some ent or | cor with sec Pro | n use curity ovide curity orma | e nicatio ers on issue some awar tion to | ès è renes | S | is n Use mit of a | nana ers c igati | age can on a curit | t that securi d correctly. perform so actions in c ty incident t m. | me ase |
| | SDLC Phase | | | | | | | s U | sed | | | | |
| Analysis Design Development Testing Deployment Main | | | | | | | Maintenar | nce | | | | | |
| N/A | Ν | /A | ١ | J/A | | N/ | Ά | | Ν | /A | | х | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester | .) | Proj | ject | : Ma | nage | er | | |
| Related Practices | | | dologie S | es, | l | Best | | | | | | | |
| Related Te | echn | ologi | es (i.e. | Jav | va) | | | | | | | | |
| | | Proje | ect 1 | | Х | Proje | ect 5 | | | Х | Pr | oject 9 | x |
| Projects Using This | | | | | | Proje | ect 6 | | | Х | Pr | oject 10 | |
| Best | | | | | х | Proje | ect 7 | | | | Pr | oject 11 | |
| | | Proje | ect 4 | | | Proje | ect 8 | | | | Pr | oject 12 | |

| BP29. B | est F | racti | ce Nan | ne: | | Pr | o-Ac | tive | e Us | ser Ir | nfoi | mation | |
|----------------------------|-------------|---------|--------------|-------|--------|--------------------------|--------|------|------|---------------|------|--------------------------------|------|
| Use | | | | Ob | jecti | ves | | | Ве | nefit | s | | |
| Incident m Release m | • | | | abo | out se | end-u ervice tions | | | the | emse | lves | djust s towards ruptions | |
| | SDLC Phase | | | | | | e It I | s U | sec | | | | |
| Analysis Design Developmer | | | | | | Test | ting | D | eplo | yme | nt | Maintenan | се |
| N/A | /A N/A N/A | | | | | N/ | Ά | | | Х | | Х | |
| Roles (i.e. | Ana | lyst, c | levelop | er, t | ester |) | Оре | erat | ions | 3 | | | |
| Related Practices | | | dologie S | es, | E | Best | | _ | | Incid nana | | manageme nent | ent, |
| Related To | echn | ologi | es (i.e. | Jav | a) | | | | | | | | |
| | | Droi | o ot 1 | | v | Droi | not F | | | Х | Dr | aiaat 0 | |
| Project | Project 1 X | | | | | Pioje | ect 5 | | | ^ | PI | oject 9 | |
| Using This | | | | | Х | Proje | ect 6 | | | | Pr | oject 10 | |
| Best Practice Project 3 | | | | | | Proje | ect 7 | | X Pi | | Pr | oject 11 | х |
| | | Proj | ect 4 | | | Proje | ect 8 | | | Х | Pr | oject 12 | Х |

| BP30. Best Pi | ractice Nam | e: (| Cont | tinuo | us Delivery | | | | |
|---------------------------------------|---|--|--------------------------------|--------|--|-----------|-----------|----------------------|----|
| Use | Obj | ectives | 5 | | Benefits | | | | |
| Release manage | faste freq allov incre upda appl | ease so er and uently l wing m ementa ates to ication luction | more by ore al the | | Ensure relia deployed a Reduces th | t any tin | ne. | which can t ırket |)e |
| | SI | DLC PI | hase | e Whe | ere It Is Use | d | | | |
| Analysis | Deve | elopment Testing Deployment Mainter | | | | | Maintenar | nce | |
| N/A | | | | | | Х | | Х | |
| Roles (i.e. A tester) | nalyst, de | velope | r, (| Opera | ations | | | | |
| Related Meth Practices and T | iodologies, ools | Bes | st (| Contir | nuous deploy | yment | | | |
| Related Techno | logies (i.e. 、 | Java) | | | | | | | |
| | Project 1 | | х | Proj | ect 5 | | Proje | ect 9 | x |
| Projects Using Project 2 This Best | | | | Proj | ect 6 | х | Proje | ect 10 | |
| Practice Project 3 | | | | Proj | roject 7 | | Proje | ect 11 | Х |
| | Project 4 | | | | Project 8 Project 12 > | | | Х | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

3.3. Tools Used by the Analysed Projects During the Software Development Lifecycle

| T1 Too | Name | : | | Ε | MC Docu | men | tum | хСР | | | |
|--------------------------|-----------|---|-------|----|---|-------|------|-------|---------|------|----|
| Use | | Objectives | | | Benefits | | | Li | censing | Тур | е |
| Development platform | | A flexible development platform for automating complex, information- intensive processes to better busines decisions | | | It improve productiv intelligen agility. | ′ity, | nd | Co | ommerci | al | |
| | | SDLC Pha | ase V | Nh | ere It Is U | Jsed | | | | | |
| Analysis | | Design | De | ve | lopment | | Test | ing | Deplo | byme | nt |
| N/A | | Х | | | Х | | N// | 4 | Ν | I/A | |
| Roles (i.e. Ar | nalyst, c | developer, teste | er) | | IT Tean | n | | | | | |
| Related Met and Tools | hodolo | ogies, Best Pr | actic | es | | | | | | | |
| Related Tecl | nnologi | ies (i.e. Java) | | | Docum | entu | m | | | | |
| | | | | | | | | | _ | | |
| | P | Project 1 | | Pı | oject 5 | | | Proje | ect 9 | | |
| Projects Us | ng | roject 2 | | Pı | oject 6 | | | Proje | ect 10 | | |
| This Tool | Р | Project 3 | | Pı | oject 7 | | | Proje | ect 11 | | |
| | Р | Project 4 | | Pı | oject 8 | | Х | Proje | ect 12 | | |

| T2 Tool Na | me: | | Jir | а | | | | | | |
|---|--|-------------|-------------|---|-------------------------|---------------------|--------------|------------|------|--|
| Use | Objectives | | | Benefits | | | Lic | ensing Ty | /pe | |
| Bug tracking system Project management software | It provides bu tracking, issu tracking, and project management functions. | e | t t i | Connection the develop that it use to the sing of truth step is projects. | loper es, m gle s | too akin ourc | ls g æ | assian | | |
| | SDLC Pł | | | ere It Is L | Jsed | | | | | |
| Analysis | Analysis Design | | | pment | - | Test | ing | Deploym | nent | |
| Х | Х | | > | < | | Х | | Х | | |
| Roles (i.e. IT Te | am, user) | | | IT Tean | n | | | | | |
| Related Method and Tools | lologies, Best Pı | racti | ces | PM2 an | id de | velo | oment i | methodolo | gy | |
| Related Techno | logies (i.e. Java) | | | JAVA | | | | | | |
| | Project 1 | Project 1 X | | | X | | Projec | xt 9 | x | |
| Projects Using | ts Using Project 2 | | | ject 6 | | Х | Projec | ct 10 | Х | |
| This Tool | х | Pro | ject 7 | | X Proje | | ct 11 | Х | | |
| | Project 3 Project 4 | | Pro | ject 8 | | Х | Projec | Project 12 | | |

| T3 Tool Na | me: | | Ya | mmer | | | | | |
|--|---|----------------------------|----------|--|--------------------------|--------------|--------|-----------|------|
| Use | Objectives | | | Benefits | | | Lice | ensing Ty | ре |
| Private communications within organizations | It is a freemin enterprise so networking so used for priva communication within organizations | cial ervic ate on | e d I | It helps e collabora departme locations business | ite ao ents, , and | cross d | Mic | rosoft | |
| | SDLC Pha | | | | se Where It Is Used | | | | |
| Analysis | Design | De | evelo | pment | | Testi | ng | Deployr | nent |
| N/A | N/A | | N/ | /A | | N/A | A | N/A | 1 |
| Roles (i.e. IT Te | am, user) | | | All stak | ehol | ders | | | |
| Related Method and Tools | lologies, Best Pi | racti | ces | Deployr | ment | t com | munica | tion | |
| Related Techno | logies (i.e. Java) | | | | | | | | |
| | Project 1 | Х | Pro | ject 5 | | | Projec | t 9 | Х |
| Projects Using | Projects Using Project 2 | | | ject 6 | | | Projec | t 10 | |
| This Tool | | | | ject 7 | | | Projec | t 11 | |
| | Project 4 | | | ject 8 | | X Project 12 | | | х |

| T4 Tool Na | me: | [| Document | um | | | | |
|--|---|--------|---|--|-------|-------|-----------|-----|
| Use | Objectives | | Benefits | 5 | | Lic | ensing Ty | ре |
| Enterprise conte management platform | nt It provides management function capabilities fo types of conte | or all | Controls the repo improve complian through compreh authenti authoris auditing | sitory ar s nce nensive cation, ation and | d | Cor | nmercial | |
| | SDLC Ph | ase W | /here It Is | Used | | | | |
| Analysis | Design | Deve | elopment | Те | sting | | Deploym | ent |
| Х | х | | х | | Х | | х | |
| Roles (i.e. IT Te | am, user) | | IT Tea | m | | | | |
| Related Method and Tools | lologies, Best Pi | actice | es | | | | | |
| Related Techno | ologies (i.e. Java) | | | | | | | |
| | Project 1 | F | Project 5 | | P | rojec | xt 9 | |
| Projects Using | Project 2 | F | Project 6 | | P | rojec | xt 10 | |
| This Tool | Project 3 | F | Project 7 | | P | rojec | xt 11 | |
| | Project 4 | F | Project 8 | Х | P | rojec | xt 12 | |

| Т5 | Tool Nan | ne: | | Cr | owd | | | | | |
|-------------------|-----------------------|---|------|-----------|----------------------------------|--------------|------------|-------|-----------|------|
| Use | | Objectives | | | Benefits | | | Lice | ensing Ty | ре |
| Identity Manag | / ement too | It provides I. sign on and identity capat | l us | er | Can be with sev repositori | veral us | ed ser | Atla | issian | |
| | SDLC Pha | | | | ere It Is l | Jsed | | | | |
| An | Analysis Design | | | | opment | pment Testir | | | Deploym | nent |
| 1 | N/A | N/A | | N | /A | N | /A | | Х | |
| Roles | (i.e. IT Tea | am, user) | | | End-Us | er, IT Te | eam | | | |
| Relate and To | | ologies, Best Pr | acti | ces | Single S | Sign On | | | | |
| Relate | d Technol | ogies (i.e. Java) | | | | | | | | |
| | | Project 1 | х | Pro | oject 5 | | Proje | | rt 9 | |
| Projec | jects Using Project 2 | | | Pro | oject 6 | | Ρ | rojec | :t 10 | |
| | This Tool Project 3 | | | Project 7 | | | Ρ | rojec | :t 11 | |
| | | Project 4 | | Pro | oject 8 | | Project 12 | | | |

| Т6 | Tool Nan | ne: | | G | itHub | | | | | |
|-------------------------------|-------------------------|--|---------------|-------------|---|---------------------------|-----------------------------|-----------------------|-------------|-----|
| Use | | Objectives | | | Benefits | | | Lic | ensing Ty | ре |
| Web-ba reposito service | sed G ry hostir | Git Distributed rong control and code manage functionality of the second | souro jeme | ce nt | It provide repositor develope push and changes from that | y wh ers d pu to | ere a ca II the an | all in ir id | veral Licen | ses |
| | SDLC Pha | | | Wł | nere It Is I | Used | k | | | |
| Ana | Analysis Design | | | Development | | | Test | ing | Deployment | |
| N | /A | N/A | | | Х | | Х | | Х | |
| Roles (i | .e. IT Tea | ım, user) | | | | | | | | |
| Related and Too | | ologies, Best P | ractio | ces | N/A | | | | | |
| Related | Technol | ogies (i.e. Java) | | | RUBY | | | | | |
| | | Project 1 | X | Pr | oject 5 | | F | | xt 9 | |
| Project | rojects Using Project 2 | | | Pr | oject 6 | | | Projec | xt 10 | х |
| This | This Tool Project 3 | | | Pr | oject 7 | 7 X | | Projec | xt 11 | |
| | Project 4 | | | Pr | oject 8 | X Projec | | | xt 12 | |

| T 7 | Tool Nan | ne: | | Co | onfluence |) | | | | | |
|---------------------------|-----------------|--|-------------|-------------|--|---------------------|---------------|--------|--------------|------|--|
| Use | | Objectives | | | Benefits | | | Lic | ensing | Туре | |
| Team collabo softwa | | Organize wor create docum and discussion board in one | nents on | в, е | It has bee adapted t with Jira a Atlassian such as E | to we and Sof | othe tware | r | assian | | |
| | SDLC Pha | | | Wh | ere It Is L | Jsec | ł | | | | |
| An | Analysis Design | | | Development | | | Test | ing | Deployment | | |
| | Х | Х | | | Х | | Х | , | > | (| |
| Roles | (i.e. IT Tea | am, user) | | | | | | | | | |
| Relate and To | | ologies, Best Pi | racti | ces | JIRA, B | AME | 300 | | | | |
| Relate | d Technol | ogies (i.e. Java) | | | JAVA | | | | | | |
| | | Project 1 | x | Pro | oject 5 | | х | Projec | ct 9 | | |
| Projec | ts Using | Project 2 | ect 2 X | | | | | Projec | ct 10 | Х | |
| Thi | s Tool | Project 3 | | | Project 7 | | | Projec | Project 11 | | |
| | Project 4 | | | Pro | oject 8 | | | Projec | Project 12 X | | |

| T8 Tool N | ame: | | sv | N | | | | | |
|---------------------------|---------------------------|---|------------------------|---|----------------------|-------------|---------|-----------|------|
| Use | Objecti | ves | | Benefits | | | Lice | ensing Ty | pe |
| Code repositor | manage and bra | used to e versions nches of rce code | (1 1 1 | It provide commits, flexible update/co and it eas integratio | fast omm se of | and its, | Ара | iche 2.0 | |
| | SDLC Pha | | | ere It Is l | Jsed | | | | |
| Analysis | Analysis Design | | | pment | | Test | ing | Deployn | nent |
| N/A | N/A | | N | /A | | Х | | Х | |
| Roles (i.e. IT | 「eam, user) | | | Develop | ber, ⁻ | Teste | ər | | |
| Related Meth and Tools | odologies, Be | est Practi | ces | Source | code | e ver | sioning | | |
| Related Tech | nologies (i.e. | Java) | | | | | | | |
| | Project 1 | X | Pro | ject 5 | XI | | Projec | rt 9 | |
| Projects Usin | ojects Using Project 2 | | | Project 6 | | XF | | rt 10 | х |
| This Tool | This Tool Project 3 > | | | ject 7 | | | Projec | :t 11 | |
| Project 4 | | | Pro | ject 8 | | Project 12 | | | Х |

| Т9 | Tool Nan | ne: | | Ne | xus | | | | | |
|------------------|---|-------------------|-------|------|--|-----------------|-------------|----------------|----------|-----|
| Use | | Objectives | | | Benefits | ; | | Licensing Type | | |
| | Libraries Manages repository used with developm projects a monitors t usage, ins security a license iss affecting t componer | | | | It provide essential infrastruc compone software developn continuo | cture ent-ba | ased and | | mmercial | |
| | SDLC Phase | | | | ere It Is l | Jsed | | | | |
| Analys | is | Design | De | velo | pment | Testing | | | Deploym | ent |
| 1 | N/A | N/A | | | x | | Х | | Х | |
| Roles | (i.e. IT Tea | am, user) | | | Develo | per, a | archit | tect, op | erations | |
| Relate and To | | ologies, Best Pı | racti | ces | | | | | | |
| Relate | d Technol | ogies (i.e. Java) | | | | | | | | |
| | Project 1 X | | | | oject 5 | | | Projec | ct 9 | |
| Projec | Projects Using Project 2 X | | | Pro | oject 6 | | Х | Projec | ct 10 | |
| | This Tool Project 3 | | | Pro | oject 7 | ect 7 | | Projec | ct 11 | |
| | Project 4 | | | Pro | oject 8 | | Projec | Project 12 | | |

| T10 | Tool Nam | e: | | Cr | uiseCon | trol | | | | |
|------------------|--|------------------|-------|-------|--|------------|----------|-----------------|----------|--------|
| Use | | Objectives | | | Benefits | | | Licensing Type | | |
| | Continuous Automates builds tests, and release | | | S . | Deploym automatio Test resu reporting | on ılts | | BSD-style licen | | |
| | SDLC Pha | | | | here It Is | : Use | ed | | | |
| Analys | Analysis Design D | | | /eloj | oment | Testing | | Deploy | | ent |
| 1 | N/A N/A | | | N | /A | | Х | | Х | |
| Roles | (i.e. IT Tea | ım, user) | | | Develo | per, t | ester, I | Т оре | erations | |
| Relate and Te | | ologies, Best Pr | actio | ces | Continu Bambo | | | | , | nkins, |
| Relate | ed Technol | ogies (i.e. Java |) | | JAVA | | | | | |
| | | Project 1 | | Pro | oject 5 | | P | rojec | t 9 | |
| | ojects ng This - | Project 2 | | Pro | oject 6 | | P | rojec | t 10 | |
| | | Project 3 | Х | Pro | oject 7 | | P | rojec | t 11 | |
| | | Project 4 | | Pro | oject 8 | | P | rojec | t 12 | |

| T11 | Tool Nan | ne: | | Je | nkins | | | | | | |
|-------------------|--------------|------------------------|--------|-------|--|------------|-------|------------|-----------------------------|--------|--|
| Use | | Objectives | \$ | | Benefits | | | Lice | Licensing Type | | |
| Contin integra | | Automates tests, and r | | es i | Deploym automatio Test resu reporting | on Ilts | | | Mit and Creative Commons | | |
| | | SDLC | Phas | e Wh | Where It Is Used | | | | | | |
| Analys | is | Design | De | velop | oment | Tes | sting | | Deploym | ient | |
| 1 | N/A | N/A | | N | N/A | | Х | | Х | | |
| Roles | (i.e. IT Tea | am, user) | user) | | | per, t | teste | r, IT op | IT operations | | |
| Relate and To | | ologies, Best | Practi | ces | Continu Bambo | | - | | Cruise co GitHub | ntrol, | |
| Relate | d Technol | ogies (i.e. Jav | a) | | JAVA | | | | | | |
| | | Project 1 | | Pro | oject 5 | | | Projec | t 9 | | |
| Projec | ts Using | Project 2 | x | Pro | oject 6 | | х | Projec | t 10 | | |
| | s Tool | Project 3 | Х | Pro | oject 7 | | | Projec | t 11 | | |
| | | Project 4 | Х | Pro | oject 8 | | | Project 12 | | х | |

| T12 | Tool Nar | ne: | | Ва | mboo | | | | | | |
|-----------------|---|--------------------|--------|-------|--|--------|-------|----------------------|-----------------------|-------------|--|
| Use | | Objectives | | | Benefits | | | Lice | Licensing Type | | |
| integra | Continuous integration server and delivery tool. Automate tests, and in a single workflow. | | | es a | Supports build any program language. | | | | ssian | | |
| | SDLC | | | | ere It Is | Use | d | | | | |
| Analys | sis | Design | De | velop | oment | Tes | sting | | Deploymer | | |
| | N/A | N/A | | N | /A | | Х | | Х | | |
| Roles | (i.e. IT Te | am, user) | | | Develop | per, t | teste | r, IT op | erations | | |
| Relate and T | | lologies, Best F | Practi | ces | Continu Cruise | | | itegratio exus, S | on, Jen SVN, GitHu | kins, Jb | |
| Relate | ed Techno | ologies (i.e. Java | a) | | Java. | | | | | | |
| | | Project 1 | X | Pro | oject 5 | | Proj | | rt 9 | | |
| | ojects | Project 2 | | Pro | oject 6 | | | Projec | t 10 | | |
| | ng This Fool | Project 3 | | Pro | oject 7 | | Х | Projec | :t 11 | | |
| | | Project 4 | | Pro | oject 8 | | | Projec | t 12 | | |

| T13 | | | | | M Ration | al | | | | |
|-------------------|---|-----------------|------|----------|---|--|----------------|----------------------------------|--------------------------|---|
| Use | | | | Benefits | | | Lice | ensing Ty | /pe | |
| Manage | Lifecycle development Management Solution SDLC Phas | | | | Provides to deliver requirem managen quality managen change a configura managen project pl and track capabilitie single pla | ents nent, nent, and ation nent a lanning cing es on | ind g | Con | nmercial | |
| | SDLC Phas | | | | ase Where It Is L | | | | | |
| Analysis | Analysis Design De | | | | lopment Te | | ing | | Deployme | |
|) | x | Х | | > | < | | Х | | Х | |
| Roles (| i.e. IT Tea | m, user) | | | IT Tean | n | | | | |
| Related and To | | logies, Best Pr | acti | ces | Applica software require manage configu project | e c ments ement ration | deve m , | lopmer anager cha manag | ment, q inge ement | ment, cycle, uality and and |
| Related | ed Technologies (i.e. Java) | | | | | | | | | |
| | Project 1 | | | Pro | ject 5 | | F | | t 9 | |
| | Projects Project 2 | | | Pro | ject 6 | | F | | t 10 | |
| | Using This Tool Project 3 | | | Pro | ject 7 | ject 7 | | Projec | t 11 | x |
| | Project 4 | | | Pro | oject 8 | | | Project 12 | | |

| T14 | Tool Nar | ne: | | Se | lenium | | | | | |
|-----------------|---|--------------------|-------|-----|---------------------|-------|--------|--------|-----------|-------|
| Use | | Objectives | | | Benefits | | | Lice | ensing Ty | /pe |
| frame | Software testing framework for web applications SDLC P | | | | Automate | ed te | sting | Ара | iche 2.0 | |
| | SDLC Ph | | | | ere It Is | Use | d | | | |
| Analys | Analysis Design | | | | oment | Tes | sting | | Deployn | nent |
| | N/A N/A | | | N | /A | | Х | | N/A | |
| Roles | (i.e. IT Te | am, user) | | | Develop | ber, | teste | r | | |
| Relate and T | | lologies, Best P | racti | ces | Functio test aut | | | ng, re | gression | test, |
| Relate | ed Techno | ologies (i.e. Java | I) | | Web Te | echn | ologie | es | | |
| | | Project 1 | | Pro | oject 5 | | | Projec | t 9 | |
| | ojects | Project 2 | х | Pro | oject 6 | | | Projec | t 10 | |
| | ng This Tool | Project 3 | | Pro | oject 7 | | | Projec | tt 11 | Х |
| | Project 4 | | | Pro | oject 8 | | Х | Projec | t 12 | х |

| T15 Tool Nai | ne: | | Cr | ucible | | | | |
|-----------------------------|--|-------|------|-------------|---|------------------|-----------|-----|
| Use | Objectives | | | Benefits | | Lic | ensing Ty | ре |
| Peer code reviev | Peer code review It provides a particularly tailore to distribution teams and facilitates asynchronous review and commenting on code SDLC Phase | | | | de review hts the f the code is revised othe rs or by eam | e e l r | prietary | |
| | SDLC Pr | nase | Wh | ere It Is l | Jsed | | | |
| Analysis | Design | De | velo | pment | Testing | | Deploym | ent |
| N/A | N/A | | х | | | | х | |
| Roles (i.e. IT Te | am, user) | | | Develo | per, QA | | | |
| Related Method and Tools | lologies, Best P | racti | ces | Code re | eview | | | |
| Related Techno | logies (i.e. Java) | | | | | | | |
| | Project 1 | | Pr | oject 5 | | Projec | xt 9 | |
| Projects Using Project 2 X | | | Pr | oject 6 | Projec | | xt 10 | |
| This Tool | | | | oject 7 | | Projec | xt 11 | |
| | Project 4 | | | oject 8 | | Project 12 | | |

| T16 | Tool Nan | ne: | | С | A Clarity | | | | | |
|-------------------|--------------------------|-----------------------------|----------------|-----------|---|---------------------------|--------------------------|-----------------------------|-----------|------|
| Use | | Objectives | | | Benefits | | | Lic | ensing Ty | /pe |
| Project manage | | related aspectors, planning | ng an proje | ke Ind | Projects desired line wit needs business strategies Manage financial your por | resu h r s. aspe | nark ar ar ects | in et ad all of | mmercial | |
| | SDLC | | | | accounta | | | | | |
| | SDLC Phase | | | | ere It Is l | Jsed | | | | |
| Analysi | s | Design | De | /elo | pment | Tes | sting | | Deploym | nent |
| Х | | Х | х | | | х | | | х | |
| Roles (| (i.e. IT Tea | am, user) | | | Project | man | ager | | | |
| Related and To | | ologies, Best P | ractio | ces | | | | | | |
| Related | d Technol | ogies (i.e. Java) |) | | | | | | | |
| | Project 1 | | | Pr | oject 5 | | XP | | ct 9 | |
| Project | Projects Using Project 2 | | | Pr | oject 6 | | Proje | | ct 10 | |
| | Tool | Project 3 | | Pr | oject 7 | | | Proje | ct 11 | |
| | Project 4 | | | Pr | oject 8 | | Project 12 | | | |

| T17 | Tool Nan | ne: | \$ | Sh | arePoint | : | | | | |
|------------------------|-------------------------|---|--|-----------|----------|------------------------------|-------|--------|------------|-----|
| Use | | Objectives | | | Benefits | | | Lice | ensing Typ | be |
| CMS Manag Syster | (Conte gement ns) | combines v functions white traditionally separate applications: intranet, ext content | combines various functions which are traditionally separate applications: intranet, extranet, content management, document | | | e writings publications e | | | nmercial | |
| | | hase V | Nh | ere It Is | Use | d | | | | |
| Analys | sis | Design | Deve | lop | oment | Tes | sting | | Deployme | ent |
| | Х | Х | | > | < | | Х | | х | |
| Roles | (i.e. IT Te | am, user) | | | End-use | er, I7 | ГТеа | am | | |
| Relate and Te | | lologies, Best Pr | actice | S | | | | | | |
| Relate | ed Techno | ologies (i.e. Java) |) | | | | | | | |
| | Project 1 | | | Pro | ject 5 | | х | Projec | st 9 | |
| | ojects ng This | Project 2 | F | Pro | ject 6 | | | Projec | rt 10 | |
| | ig mis ool | Project 3 | F | Pro | ject 7 | | | Projec | :t 11 | |
| | | Project 4 | Pi | | ject 8 | | | Projec | t 12 | |

| T18 Tool Na | me: | Piv | wik | | | | |
|----------------------------|--------------------|---------|------------|------------|------------|-----------|-----|
| Use | Objectives | | Benefits | | Lice | ensing Ty | ре |
| Web analytics | | orts on | | | GN | U GPL v3 | |
| | SDLC P | hase Wł | nere It Is | Used | | | |
| Analysis | Design | Develo | pment | Testing | | Deploym | ent |
| Х | х | | x | Х | | Х | |
| Roles (i.e. IT T | eam, user) | | End-us | er, IT Tea | am | | |
| Related Metho and Tools | dologies, Best Pi | actices | | | | | |
| Related Techn | ologies (i.e. Java |) | | PHP, M | lySQL | | |
| | Project 1 | Pro | oject 5 | | Projec | rt 9 | x |
| Projects | Project 2 | Pro | oject 6 | | Projec | t 10 | |
| Using This Tool | Project 3 | Pro | oject 7 | | Projec | :t 11 | |
| | Project 4 | | | | Project 12 | | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

3.4. Libraries and Building Blocks Used by the Analysed Projects During the Software Development Lifecycle

| LB&B1 | Librar | y Name: | | Alfre | sco Se | curity | | | |
|-----------------------------|-----------|--|---|-------------|--|-----------|-------------------|-----------------------------------|------|
| Use | | Objectives | | Benefits | | | Lice | ensing Ty | pe |
| Authenticat Authorisatio | | combination c authentication and | combination of authentication and Authorisation. | | An internal, password-based, authentication implementation Support to integra with many externa authentication environments The option to write your own authentication integration and to several of these options simultaneously | | | ser GNU lic Licens (LGPLv3) | e |
| | | SDLC Ph | ase \ | Where It Is | s Used | | | | |
| Analysis | | Design | Dev | elopment | Testir | ng | | Deploym | ient |
| Х | | Х | | Х | | Х | | Х | |
| Roles (i.e. | Analyst | , developer, test | er) | | | | | | |
| Related Practices a | | odologies, ols | Best | t | | | | | |
| Related Te | echnolo | ogies (i.e. Java) | | | L | DAP, J | JAVA | | |
| | Project 1 | | | Project 5 | | Pro | ject 9 |) | |
| Projects U | lsina | Project 2 | | Project 6 | | Pro | ject ² | 10 | |
| This Libr | arv | Project 3 | | Project 7 | | Pro | ject ² | 11 | |
| | | Project 4 | | Project 8 | | Project 2 | | 12 | х |

| LB&B2 | | | | | C | CA | S | | | | | | |
|--|--------------------------|-----|------------------|----------|-------|----------|---------------------|---------|------------|-------|-------------------|-----------|-----|
| Use | | | Objective | es | | | Benefits | 5 | | | Licensing Type | | |
| Multiprotocol Web single sign-on Single Sign On too that can be integrat with a number of supported authentication mechanisms including LDAP/Active Directory, Kerberos and RDBMS | | | | | rated | | Provides On capa | | | Sign | Ap | bache 2.0 | |
| | SDLC Ph | | | | | Wł | nere It Is | s Us | ed | | | | |
| Analysis | | De | esign | Develo | pmer | nt | | Tes | ting | | | Deploym | ent |
| Х | | Х | | Х | | | | Х | | | | Х | |
| Roles (| i.e. IT | Tea | am, user. |) | | | End-us | ser, l | T tea | Im | | | |
| Related Practic | | | thodolog ools | ies, | Bes | st | Single | Sign | On | | | | |
| Related | l Tech | no | logies (i.e | e. Java) | | | ECAS | | | | | | |
| | Project 1 | | | | | Ρ | roject 5 | ect 5 | | Proje | ct 9 | 9 | |
| | Projects Using Project 2 | | | | Ρ | roject 6 | ect 6 | | Project 10 | | 10 | | |
| This L | This Library Project 3 | | | | Ρ | roject 7 | | X Proje | | ct 1 | 11 | | |
| | Project 4 | | | | Ρ | roject 8 | | | Proje | ct 1 | 12 | | |

| LB&B31 Tool Name: | | | Е | ECAS | | | | | | | |
|--|------|---|-------------------------|-------------|-------------------|---------------|-----|--------|----------------|------|--|
| Use | | Objectives | Objectives | | | Benefits | | | Licensing Type | | |
| European Institutions Authorisation system based CAS | on | Single Sign o that can be integrated to number of supported authentication mechanisms including LDAP/Active Directory, Kerberos, and RDBMS | sign On capabilities | | | gle | EU | ΡL | | | |
| | | SDLC Pha | ase | Wh | ere It Is U | lsed | | | | | |
| Analysis | D | esign | De | velo | opment | oment Testing | | | Deploym | nent | |
| N/A | | N/A | | 1 | V/A | | N/. | A | Х | | |
| Roles (i.e. IT Tea | m, 1 | user) | | | End-user, IT Team | | | | | | |
| Related Method and Tools | olo | gies, Best Pr | acti | ces | Single | Sign | On | | | | |
| Related Technol | ogie | es (i.e. Java) | | | CAS, Ja | ava | | | | | |
| | Р | roject 1 | | Project 5 | | | x | Projec | et 9 | X | |
| Projects Using | Р | roject 2 | х | X Project 6 | | | | Projec | t 10 | x | |
| This Library | Р | roject 3 | х | K Project 7 | | | Х | Projec | t 11 | х | |
| | Ρ | roject 4 | х | Pı | roject 8 | | | Projec | t 12 | х | |

| LB&B4 Tool Nan | ne: | E | E-Signature | | | | | | |
|------------------------------|--|---------|---|----------------|------------|------------|--------------|----|--|
| Use | Objectives | | Benefits | ; | | Lice | ensing Ty | ре | |
| Electronic Authentication | Demonstrate authenticity digital messa document | | Provides a level of assurance that the message or document was created by a known person | | | N/A | | | |
| | SDLC Ph | ase Wł | nere It Is L | Jsed | | | | | |
| Analysis | Design | Deve | lopment | opment Testing | | | g Deployment | | |
| N/A | N/A | | N/A | N | /A | | N/A | | |
| Roles (i.e. IT Tear | m, user) | | End-Us | End-User | | | | | |
| Related Methodo and Tools | ologies, Best Pr | actices | 5 N/A | | | | | | |
| Related Technolo | ogies (i.e. Java) | | N/A | | | | | | |
| | Project 1 | P | Project 5 | | | Project 9 | | | |
| Projects Using | Project 2 | Р | | Pr | Project 10 | | | | |
| This Library | Project 3 | Р | roject 7 | | Pr | Project 11 | | | |
| | Project 4 | Р | roject 8 | X | Pr | ojec | :t 12 | | |

| LB&B5 Library Name: | | | OpenSSL | | | | | | | |
|----------------------------|---|---|-------------------------------|----------------------------|------|-------|-------------|---|---|--|
| Use Objectives | | | Benefits | | | | Lic | Licensing Type | | |
| Security Libra | ry Software like be used applications need to communicat against eavesdroppi need to as the identity party at the end | i i tha secur ions ng c scertai of th | n at re or n e | cryptographic functions | | | 1.0 clau | Apache license 1.0 and four clause BSE License | | |
| | SDLC P | hase | Wh | ere It Is L | Jsed | | | | | |
| Analysis | Design | Dev | elo | pment | Tes | sting | | Deployment | | |
| N/A | N/A | | Ν | I/A | | N/ | A | X | | |
| Roles (i.e. Anal | yst, developer, tes | ter) | | | | | | | | |
| Related Metho and Tools | dologies, Best F | Practic | es | Encrypt | tion | | | | | |
| Related Techn | ologies (i.e. Java) |) | | | | | | | | |
| | Project 1 | | Project 5 P | | | | | ct 9 | | |
| Projects Using | Project 2 | | Project 6 | | | Х | Projec | Project 10 | | |
| This Library | Project 3 | | Project 7 | | | | Projec | ct 11 | Х | |
| | Project 4 | | Pr | oject 8 | | | Projec | ct 12 | | |

| LB&B6 | Tool Nam | ie: | | Site Minder | | | | | | |
|-----------------------------|-----------------------------|---------------------------------------|-----------|-------------|------------------------------------|--------|--------|------------|----------------|------|
| Use | Use Objectives | | | Benefits | | | | Lic | Licensing Type | |
| | ithenticatio gle sign-on | · · · · · · · · · · · · · · · · · · · | also t | | Authentication with Screen Name | | h Cor | Commercial | | |
| SDLC Phase Where It Is Used | | | | | | | | | | |
| Ana | llysis | Design | Dev | elo | opment Testing | | | ng | Deploym | nent |
| N | /A | N/A | | N/ | Ά | | N/A | 4 | Х | |
| Roles (i | .e. IT Tear | n, user) | | | End-Us | er, IT | Tea | Im | | |
| Related and Too | | ologies, Best Pr | actice | es | Single S | Sign (| On | | | |
| Related | Technolo | ogies (i.e. Java) | | | | | | | | |
| | | Project 1 | F | Project 5 | | | | Projec | ct 9 | |
| Project | s Using | Project 2 | Project 6 | | | | Projec | ct 10 | | |
| | ibrary | Project 3 | F | Project 7 | | | | Projec | Project 11 | |
| | | Project 4 | F | ⊃ro | ject 8 | | Х | Projec | ct 12 | |

| LB&B7 Libr | ary Nar | ne: | | S | oring Sec | urit | y | | | |
|---|----------------|------------------|-------------|-----------------|--|-------|--------|----------|----------------|-----|
| Use | Use Objectives | | | | Benefits | | | | Licensing Type | |
| Authenticatio access contro framework | | | | | Provides authentication, authorisation and other security features for enterprise applications | | | Ара | Apache 2.0 | |
| | | SDLC Pha | ase | Wh | ere It Is L | lsed | | | | |
| Analysis | D | esign | De | Development Tes | | | sting | | Deploym | ent |
| N/A | | Х | | | х | | Х | | Х | |
| Roles (i.e. Ar | nalyst, c | leveloper, teste | er) | | End-user, IT team | | | | | |
| Related Met and Tools | hodolo | gies, Best Pr | acti | ces | Authori | satic | on, au | Ithentic | ation | |
| Related Tecl | nnologi | es (i.e. Java) | | | JAVA/J | AVA | EE | | | |
| | Р | roject 1 | | Pr | oject 5 | | | Projec | ct 9 | |
| Projects Us | ing P | roject 2 | X Project 6 | | | | Х | Projec | ct 10 | |
| This Libra | r y P | roject 3 | | Pr | oject 7 | | | Projec | ct 11 | |
| | Р | roject 4 | | Pr | oject 8 | | | Projec | ct 12 | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

3.5. Programming languages used by the analysed projects for software development

| LG1 Language Name: | | | Ja | Java | | | | | | | |
|--------------------|-----------------|--|---|----------|---------------------------------------|---------|------------------------|----------------|------------|----|--|
| Use Objectives | | | | Benefits | | | Lice | Licensing Type | | | |
| | 0 | Provides concurrent, c based, object oriented, and specifically designed to h few implement dependencies possible. | es Java code can run on all platforms that support Java ed, and is without the need cally for recompilation plementation dencies as | | | /a d | Oracle America INC. | | | | |
| | | SDLC Pr | nase | Wh | ere It Is l | Jsed | | | | | |
| Analys | is | Design | De | velop | pment | Testir | ng | | Deploymen | | |
| 1 | N/A | Х | |) | x | | Х | | Х | | |
| Roles | (i.e. Analys | st, developer, test | ter) | | Developer, tester, analyst, architect | | | | | ct | |
| Relate and To | | ologies, Best Pı | actio | ces | Object | Oriente | ed | | | | |
| Relate | d Technol | ogies (i.e. Java) | | | | | | | | | |
| | | Project 1 | x | Pro | Project 5 | | | Projec | t 9 | | |
| | ts Using his | Project 2 | х | Pro | oject 6 | X | (| Projec | t 10 | х | |
| | amming guage | Project 3 | х | Pro | oject 7 | | | Projec | t 11 | х | |
| | | Project 4 | | Pro | oject 8 | X | (| Projec | Project 12 | | |

| LG2 | Languag | e١ | lame: | | Ы | PHP | | | | | | | |
|--------------------------------------|-----------------|-------|--|--------|-----------|----------|---|--------|-------|----------------|------------|-----------|------|
| Use | Use Objectives | | | | | Benefits | | | | Licensing Type | | | |
| Genera softwa progra langua | mming | | Programming language orig designed to o web sites | jinall | | р | Flexible and powerful specially to create web sites | | | | РНР | | |
| | | | SDLC Ph | ase | Wł | ne | re It Is l | Jsec | | | | | |
| Analys | is | D | esign | De | velc | р | pment Testing | | | | | Deploymer | |
| 1 | N/A | | Х | | | Х | x x | | | | Х | | |
| Roles | (i.e. Analy | st, (| developer, test | er) | | | Develop | ber, t | teste | r, a | naly | st, archi | tect |
| Relate and To | | olo | ogies, Best Pr | acti | ces | 3 | Object (| Orie | nted | | | | |
| Relate | d Techno | log | ies (i.e. Java) | | | | | | | | | | |
| | | Pı | roject 1 | | Pr | roj | ect 5 | | | Pr | ojec | t 9 | x |
| | ts Using his | Pı | roject 2 | | Project 6 | | | | | Pr | ojec | t 10 | |
| | amming guage | Pı | roject 3 | | Project 7 | | | | Х | Pr | Project 11 | | |
| | | Pı | roject 4 | х | Pr | roj | ect 8 | | | Pr | Project 12 | | |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4. Analysis of identified software development methodologies used in the European Institutions

Deliverable 1 provided a list of 15 projects to be analysed. Out of the 15 projects, 11 were from the European Commission and the remaining 4 from the European Parliament. The information gathering was conducted in interviews and a total of 14 out of 15 projects were covered during the interview rounds; thus, the analysis is based on the responses from the interviewees of these projects.

4.1.Project Management

4.1.1.Methodologies

This point shows the different methodologies used for project management (based on the 14 analysed projects).

Table 4-1 Analysis of identified software development methodologies used in the European Institutions – Project management methodology

| Project Management methodology | Out of the 14 analysed projects | | | | | | |
|--------------------------------|---------------------------------|----------------------------|--|--|--|--|--|
| | Number of projects | Percentage of the projects | | | | | |
| PM2 | 10 | 71% | | | | | |
| PM for EP | 3 | 21% | | | | | |
| N/A | 2 | 14% | | | | | |

The table shows that most of the projects analysed use a specific project management methodology, either PM2 or PM for EP. The relationship between the project management methodologies and the European Institutions (EC, EP) is depicted in the following figures.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

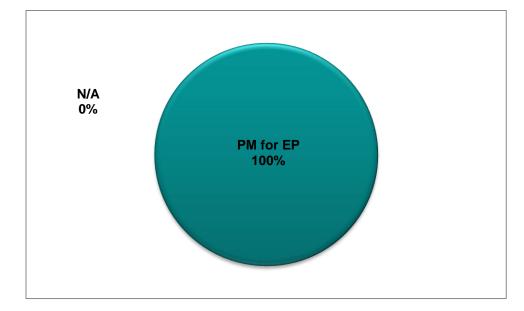


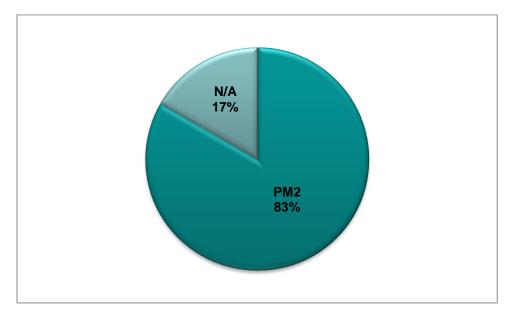
Figure 2. Analysis of identified software development methodologies used in the European Institutions – Usage of PM for EP methodology in the European Parliament

| Project Management methodology | Out of the 14 analysed projects | | | | | | |
|--------------------------------|---------------------------------|----------------------------|--|--|--|--|--|
| Tojoot managomont monouology | Number of projects | Percentage of the projects | | | | | |
| PM for EP | 3 | 100% | | | | | |
| N/A | 0 | 0% | | | | | |

This chart shows that all the projects analysed from the European Parliament use PM for EP as a project management methodology.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Figure 3. Analysis of identified software development methodologies used in the European Institutions – Usage of PM2 methodology in the European Commission



| Project Management methodology | Out of the 14 analysed projects | | | | | | |
|---------------------------------|---------------------------------|----------------------------|--|--|--|--|--|
| r rejeet management methodology | Number of projects | Percentage of the projects | | | | | |
| PM2 | 10 | 83% | | | | | |
| N/A | 2 | 17% | | | | | |

This chart shows that the majority of the European Commission projects analysed follows the PM2 methodology for project management.

4.1.2.Tools

- Jira
- Clarity

4.1.3.Conclusion

Most of the analysed European Institutions take advantage of the project management methodologies developed by them (PM2 and PM for EP). Since both methodologies are PMBOK based, they are similar in nature, and as such, they ensure that project management is well covered in the projects analysed

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.2. Software Development Lifecycle

4.2.1.Software Development Lifecycle Methodologies

4.2.1.1. Methodologies

In this point we will analyse the software development methodologies used within the analysed projects. The use of these methodologies helps the development team to follow structured processes to manage and produce high quality software products.

The following table shows the methodologies used in the projects analysed.

Table 4-2 Analysis of identified software development methodologies used in the European Institutions – Software development methodologies (Detail)

| Software Development Methodology | Out of the 14 analysed projects | | | | | | |
|----------------------------------|---------------------------------|----------------------------|--|--|--|--|--|
| | Number of projects | Percentage of the projects | | | | | |
| Scrum | 7 | 50% | | | | | |
| Custom Scrum | 5 | 36% | | | | | |
| Kanban | 3 | 21% | | | | | |
| Agile@EC | 2 | 14% | | | | | |
| Waterfall | 1 | 7% | | | | | |
| RUP@EC | 1 | 7% | | | | | |

As it is showed in the table, the majority of the projects use agile methodologies, only one project uses Waterfall and one project uses RUP@EC. Among the agile methodologies, the most widely used is Scrum, in 50% of the projects, followed by custom methodologies based on Scrum. Three projects complement Scrum with the Kanban methodology.

The previous table identifies the high level software development methodologies used by the projects of this study. In order to manage this information, they have been grouped, as shown in Figure 4.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

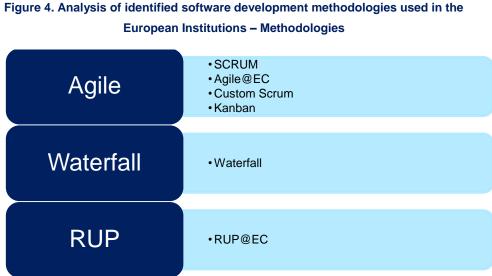


Table 4-3 Analysis of identified software development methodologies used in the European Institutions - Software development methodologies

| Software Development Methodology | Out of the 14 analysed projects | | | | | |
|----------------------------------|---------------------------------|----------------------------|--|--|--|--|
| Conware Development methodology | Number of projects | Percentage of the projects | | | | |
| Agile | 13 | 93% | | | | |
| Waterfall | 1 | 7% | | | | |
| RUP | 1 | 7% | | | | |

As it can be seen in the table, most of the projects analysed follow the agile approach, whereas only one project uses waterfall and another one uses RUP.

The following table shows the approach followed with regard to the development lifecycle according to the identified methodologies.

Table 4-4 Analysis of identified software development methodologies used in the European Institutions - Software development methodologies (cycle)

| Software development cycle | Out of the 14 analysed projects | |
|----------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Iterative | 13 | 93% |
| Waterfall | 1 | 7% |

Document elaborated in the specific context of the EU - FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 84 of 110

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Considering that RUP and Agile are iterative cycle methodologies, only one of the projects analysed uses a waterfall approach.

4.2.1.2. Tools

- Atlassian Jira
- Atlassian Confluence
- IBM Rational

4.2.1.3. Conclusion

As it has been shown in all the charts and tables analysed, the majority of the projects use an agile methodology for software development. Since agile is a methodology that delivers new functionalities in an iterative and incremental way, we can conclude that most of the projects analysed are software delivery oriented. This approach simplifies steps, formalities and artefacts, in comparison with the more procedural approaches of methodologies like Waterfall or RUP.

The analysis had to consider that some approaches are agile but not supported by applying a complete methodology like Scrum or Kanban, 36% of the projects analysed use a reduced part of these methodologies, thus, some parts of the software development lifecycle could not be well covered for these projects.

4.2.2. Security Requirements

4.2.2.1. Security Requirements Definition

Frequently software development is focused in the implementation of the functionalities which are requested by the business owners. However, from a security perspective, an attacker would be more interested on what the application will allow him to do (not necessarily related to the functionalities) and thus use it for his own benefit. Following OWASP [2] best practices, it is recommended to conduct an analysis of the security requirements in the early stages of the software development lifecycle, mainly during the design and analysis phases. This point analyses how these requirements are fulfilled in the projects analysed.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

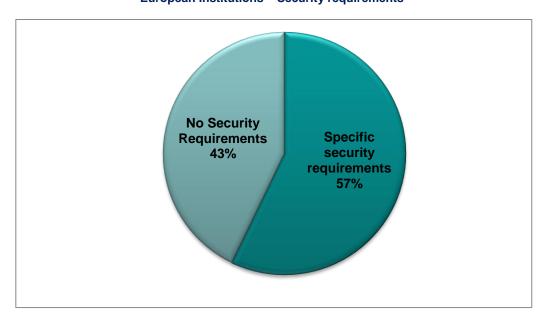


Figure 5. Analysis of identified software development methodologies used in the European Institutions – Security requirements

| Security requirements | Out of the 14 analysed projects | |
|--------------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Specific security requirements | 8 | 57% |
| No security requirements | 6 | 43% |

A bit more of the half of the analysed projects use specific security requirements gathered from a security definition phase at the beginning of the project. For the remaining projects, security requirements are defined within business requirements, so they are not explicitly addressed.

The study also shows that 43% of the projects have a security plan.

4.2.2.2. Conclusion

This section shows that only half of the projects analysed take into consideration the requirements from the security point of view. Some IT teams from analysed projects consider that the security requirements could be more relaxed if the application is not exposed to the internet, at this point it is necessary to consider that an application could be attacked not only from the internet, but also from within the intranet.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.2.3. Testing and Validation

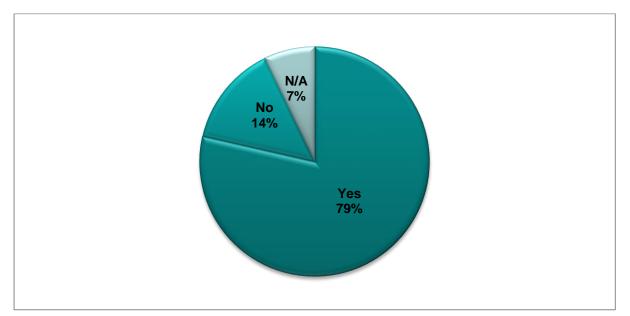
Testing is performed by DIGIT test centre in order to find possible bugs or vulnerabilities before the application is deployed in the production environment, and also to ensure the quality of the product and the implementation of the expected requirements. One main step of this phase is to receive the validation from the customer.

All the projects analysed conduct tests, but it is important to analyse how they are being conducted

4.2.3.1. Automatic Testing

Automatic testing allows the execution of written test, mostly in silent mode, without the manual intervention of the development or QA teams. Frequently this approach is based on suites or groups of individual tests which are logically-related, incremental, and repeatable. This analysis considers functional, non-functional, unit and regression tests.





| Perform automatic tests | Out of the 14 analysed projects | |
|-------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Yes | 11 | 79% |
| No | 2 | 14% |
| N/A | 1 | 7% |

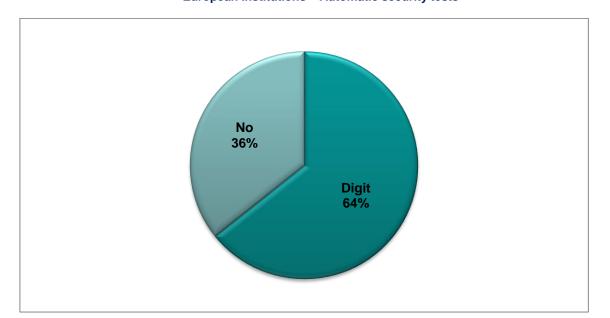
Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 87 of 110

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

The chart shows that most of the analysed projects use automatic testing for unit, functional, nonfunctional and regression test.

4.2.3.2. Security testing

This point analyses if projects execute tests to identify security bugs or vulnerabilities. These tests use techniques like penetration test, vulnerability scan and black and/or white box testing.



| Figure 7. Analysis of identified software development methodologies used in the |
|---|
| European Institutions – Automatic security tests |

| Perform automatic security tests | Out of the 14 analysed projects | |
|----------------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Digit | 9 | 64% |
| No | 5 | 36% |

As it is represented by the chart, more than half of the projects analysed use the security vulnerability testing service provided by Digit, the rest do not perform vulnerability testing for their applications.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.2.3.3. Validation Testing

Most of the projects analysed perform validation tests that are conducted by the business stakeholders in order to validate the application's functionalities. These tests are performed once the QA team validates the tests conducted by the IT team. The sequence is as follows.

Figure 8. Analysis of identified software development methodologies used in the European Institutions – Validating testing flow



The analysed projects use a special environment to perform these validation tests. These environments are analysed in section 4.2.4.1

4.2.3.4. Tools and methods

- Selenium
- JUnit
- Continuous deployment
- Security tools used by the Digit cyber-security team¹

4.2.3.5. Conclusion

This point shows that most of the analysed projects perform automatic tests. Although some tests cannot be automated, it is a good practice to perform complete and repeatable tests automatically, following the ITIL [2] "Service Validation and Testing stack" as a best practice:

- Release Component Acquisition: It is covered when the developers execute initial tests after new code is developed.
- **Release Test**: It is covered by regression, functional, non-functional, integration and unit tests which are executed within the test environment (this environment is covered in point 4.2.4.1.)

¹ Tools used by this department are not within the scope of this analysis.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

• **Service Acceptance Testing**: It is covered by the business stakeholders, who execute acceptance tests within the acceptance environment (this environment is covered in point 4.2.4.1).

Although most of the projects analysed conduct testing, not all the projects do security testing, which could help mitigate the risk of vulnerabilities

It is important to highlight that some IT teams believe that it is not necessary to perform security test when the application is deployed within the intranet, not taking into consideration the fact that attacks can be triggered by insiders.

4.2.4. Release Management.

It is the process of managing a software build through different stages and environments, including the testing (4.2.3) and deployment (4.2.4.1).

One of the main objectives of the projects that use Agile is to increase the frequency of releases to reduce the Time to Market. At this point, the concept continuous delivery influences how the transitions from development to productions are.

4.2.4.1. Deployment

Deployment is the phase where the binaries of the software are installed in the servers in order to expose the functionalities of the application.

Regarding the deployment practices preferred in the European Institutions, we should differentiate two types of environments:

- Test and Acceptance / Pre-production: Where the application is deployed to be tested and accepted by the team involved in the project (IT team and business stakeholders), before the deployment in the production environment.
- Production: Where the application is deployed to be used by end users.

In our analysis most of the projects and developments split the environments into test, acceptance/preproduction and production following the pipeline shown in Figure 9.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Figure 9. Analysis of identified software development methodologies used in the European Institutions – Delivery pipeline



Table 4-5 Analysis of identified software development methodologies used in the European Institutions – Delivery pipeline

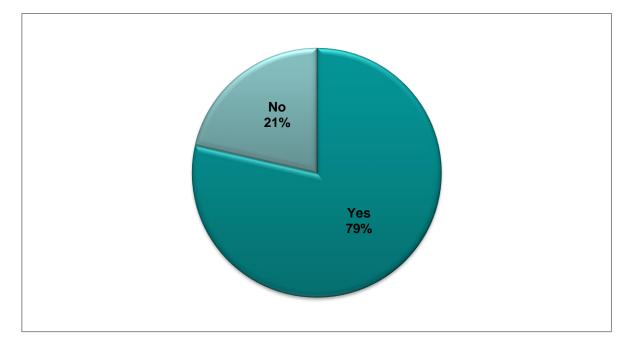
| Environment | Description |
|----------------------------------|--|
| Development | Developers use this environment to work and perform initial development tests of the application. |
| Test | Test the application to ensure the quality of the product before the deployment in the Acceptance environment. In some projects this acts as integration environment. |
| Acceptance / Pre - production | It is the environment before production. Once tests in the Test environment are successful, the version is promoted to this environment, where business stakeholders test the functionalities according to the original requirements. This environment should be as similar as possible to the production environment to mitigate the risks associated with promoting to the Production environment. |
| Production | When the acceptance/pre-production version is approved, it is promoted to this environment in order to offer the functionalities of the application to end users. This is the environment that manages the production data or "real" of the application. For the majority of the projects the promotion to this environment is performed by an operational team, commonly an external team in charge of the systems layer. |

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

To deploy and promote versions between environments, continuous deployment is a good practice and it is widely used within the projects analysed. This practice allows more frequent deployments which in turn helps to find bugs or vulnerabilities faster.

Most of the projects use continuous deployment as a trigger to execute automatic regression, integration and unit tests. Additionally, some projects use tools for automatic code review to ensure the quality of the code.

The following chart depicts the usage of continuous deployment.





| Continuous deployment | Out of the 14 analysed projects | |
|-----------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Yes | 11 | 79% |
| No | 3 | 21% |

4.2.4.2. Continuous testing and validation

Continuous testing is the process of executing automated tests as part of the delivery pipeline. The focus is on receiving continuous feedback on the business risks related to a software release candidate and determining if the software is ready to be promoted through the delivery pipeline. During this process,

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

functional and non-functional tests (4.2.3.1), static code analysis (4.2.5.1) and security testing (4.2.3.2) can be involved.

The combination of figures 6 and 10 shows that all the analysed projects which implement automatic testing also use continuous deployment, which means that they take advantage of continuous testing.

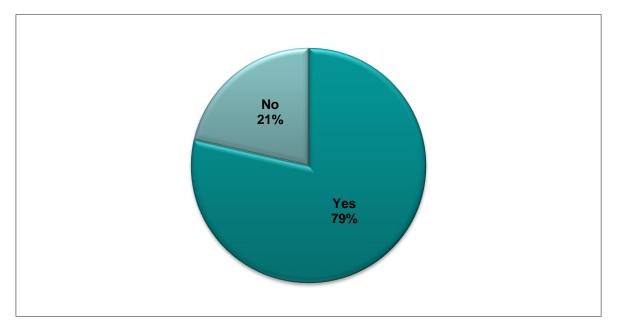
4.2.4.3. DevOps

This practice is based on agile methodologies in which the development and operational teams are involved from the beginning. This practice ensures the alignment between these two teams to mitigate the risks when the deployment is performed. Only one of the projects analysed uses this methodology since their development and operational teams are internal and work closely.

4.2.4.4. Release Planning

This point analyses if the projects use a roadmap for planning new releases. Additionally, we will analyse if the roadmap includes security aspects.





| Roadmap definition | Out of the 14 analysed projects | |
|--------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Yes | 11 | 79% |
| No | 3 | 21% |

Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 93 of 110

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

As it is depicted in the chart, 79% of the analysed projects use a roadmap to plan future features and updates. Taking into account these projects, 27% include security aspects.

4.2.4.5. Channels and tools used

All the tools used for continuous deployment are open source. The usage of each one is showed in the table below.

Table 4-6 Analysis of identified software development methodologies used in the European Institutions – Continuous delivery tools

| Continuous delivery tool | Out of the 14 analysed projects | |
|--------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Jenkins | 5 | 36% |
| Bamboo | 4 | 29% |
| Cruise Control | 2 | 14% |

The majority of the projects use Jenkins as continuous deployment tool, followed by Bamboo and only two projects use Cruise Control.

In order to communicate service interruptions derived from the deployment in production, some analysed projects communicate these interruptions to end-users using the tools listed in section 4.3.1.5.

4.2.4.6. Conclusion

This point shows that most of the analysed projects have a well-defined deployment pipeline to promote releases between environments. Additionally, most of the projects analysed take advantage of the continuous delivery best practice, which ensures the rapid identification of bugs and vulnerabilities.

There are two projects that apply the DevOps methodology. This methodology mitigates the problems between the development and operations teams when the release has to be deployed into the servers. These problems are solved by both teams working closely in all phases of the project following an agile approach.

Taking into consideration ITIL [2] Release Management as a best practice, the following processes are covered:

- Release Planning: It is covered by the roadmap for the planning of the releases.
- Release Build: It is covered by the automatic deployment performed by continuous integration servers.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

• **Release Deployment:** Most of the deployments in production are conducted by an external operations team within Digit that is in charge of the infrastructure administration.

4.2.5.Inspection and code review

4.2.5.1. Code review

This point introduces the teams in charge of performing code reviews for the analysed projects.

Table 4-7 Analysis of identified software development methodologies used in the European Institutions – Code review teams

| Code review team | Out of the 14 analysed projects | |
|---|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Development | 6 | 43% |
| Internal but different from development | 4 | 28% |
| No | 3 | 21% |
| External | 2 | 14% |
| FOSS Communities | 1 | 7% |

The majority of the projects rely on the development team for the code review, as they inspect the code produced by another developer (peer-to-peer review) or by using automatic tools like SonarQube. These are followed by projects which use different teams to perform code review. Two projects use external resources to do the code review, while another one uses the code that has been reviewed by the FOSS communities. It is worth noticing that three of the analysed projects do not conduct any code reviews.

4.2.5.1.1.Tools

- SonarQube
- Crucible
- SVN
- GitHub

4.2.5.2. Projects reviewed by security experts

This point shows who reviews the security of the projects developed.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

| Security experts team | Out of the 14 analysed projects | |
|-----------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Digit | 8 | 57% |
| External | 4 | 29% |
| No | 3 | 21% |
| Internal | 2 | 14% |

Table 4-8 Analysis of identified software development methodologies used in the European Institutions – Security experts review

Most of the projects developed by the European Commission are submitted to a service offered by Digit to review and execute automatic tests to ensure the security of the projects. 29% of the projects use specialised external services in order to check the security of the project. 14% use an internal expert and 21% do not get an expert to perform a security review.

4.2.5.3. Phase where the project is reviewed by security experts

This point shows in what phase the project is reviewed by security experts. The following table shows when the security is reviewed, in the case of those projects that perform security reviews.

Table 4-9 Analysis of identified software development methodologies used in the European Institutions – Security review phase

| Security review phase | Out of the 14 analysed projects | |
|-------------------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| At the end (test, deployment) | 6 | 54% |
| In all phases | 2 | 18% |
| At the beginning (analysis, design) | 2 | 18% |
| During development | 1 | 9% |

A good approach is to take security into consideration in all phases, with more emphasis at the beginning of the project. It is worth mentioning that 45% of the projects take security into consideration at the beginning of the project. However, the majority conduct the security review after the development phase.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.2.5.4. Conclusion

Code review ensures the quality of the code and helps to find errors which could turn into bugs or vulnerabilities in the future. A good approach is to assign the review to a team of reviewers that are not involved in the development of such code.

On the other hand, although the projects are reviewed by security experts, they concentrate on conducting only security tests. Following the recommendations from OWASP [2], a good approach is to perform a code review that also focuses on security, and to conduct security reviews during all phases of the development lifecycle, focusing on the early phases.

4.2.6. Application authentication and authorisation

This section analyses which methods of authorisation and authentication are implemented in the projects analysed.

4.2.6.1. Authentication

This point analyses the methods and tools used to conduct the authentication in the projects.

| Authentication method | Out of the 14 analysed projects | |
|-----------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| ECAS | 11 | 79% |
| CAS | 2 | 14% |
| Biometrics | 1 | 7% |
| CA Site Minder | 1 | 7% |
| Crowd | 1 | 7% |
| Others | 1 | 7% |

Table 4-10 Analysis of identified software development methodologies used in the European Institutions – Authentication methods

Most of the analysed projects use a CAS solution (since ECAS is based on CAS), one project uses advanced security systems like biometric authentication, another one uses CA Site Minder and another one use Crowd. This analysis shows that most of the applications use Single Sign On Authentication, since ECAS, CAS CA Site Minder and Crowd implement this technology.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.2.6.2. Authorisation

This point analyses how applications manage authorisations.

Table 4-11 Analysis of identified software development methodologies used in the European Institutions – Authorisation

| Authorisation | Out of the 14 analysed projects | |
|------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Based on Roles | 10 | 71% |
| Based application/tool | 3 | 21% |
| Based on Groups | 2 | 14% |
| N/A | 2 | 14% |

As it is showed in the chart most of the applications use an authorisation based on roles, few applications use groups and the rest uses the authorisation method offered by the base application or tool (i.e. Drupal, Alfresco).

4.3. Project Maintenance

In order to ensure the maintenance of the project and its sustainability, it is important to perform a good maintenance to fix bugs and vulnerabilities fast and in a preventive rather than reactive way.

4.3.1.1. Incident management support and categorisation

In this point we will analyse the tools, channels and procedures used to report bugs or vulnerabilities.

The following table analyses the entry points to report incidents.

Table 4-12 Analysis of identified software development methodologies used in the European Institutions – Bug and vulnerability reporting

| Entry Point | Out of the 14 analysed projects | |
|-------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Helpdesk | 11 | 79% |
| Jira | 6 | 43% |
| Mailbox | 1 | 7% |

Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 98 of 110

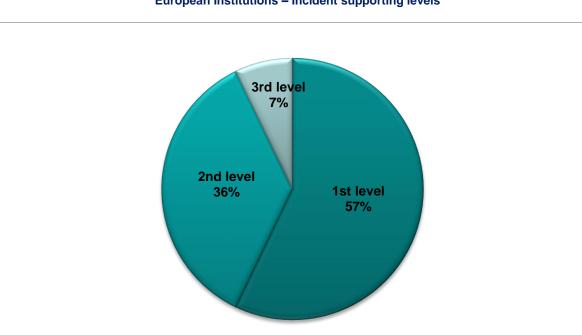
Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

As the table shows most of the analysed projects use the helpdesk as an entry point for end-users to report bugs or vulnerabilities, 43% use Jira and 7% use their mailbox. The majority of the analysed projects that implements helpdesk as an entry point use SMT as a management tool.

Helpdesk is responsible for prioritising and categorising reported bugs or vulnerabilities from the projects that have use it as an entry point. In the rest of the cases, it is the development team who prioritises and categorises.

4.3.1.2. Incident resolution

For the analysed projects that have Helpdesk, this acts as a 1st level support to solve incidents related to user operations. Some of the projects analysed use a 2nd level support which is mostly composed of technicians from the IT team; if the problem cannot be solved, it is escalated to the 3rd level, which can be the problem management team within the project or the product provider.





| Incident supporting level | Out of the 14 analysed projects | |
|---------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| 1 st level | 8 | 57% |
| 2 nd level | 5 | 36% |
| 3 rd level | 1 | 7% |

Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 99 of 110

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

• **Projects with 1st level support**: This support level is performed by the helpdesk to resolve issues raised by the end-users. If the issue is not solved, it is escalated to problem management 4.3.2.2.

Figure 13. Analysis of identified software development methodologies used in the European Institutions – 1st level support flow



• **Projects with 2nd level support**: if the issue cannot be solved at the 1st level support, it is escalated to the 2nd level, and handled by specialized technicians from the IT team. If the issue is not solved, it is escalated to problem management 4.3.2.2.

Figure 14. Analysis of identified software development methodologies used in the European Institutions – 2nd level support flow



Projects with 3rd level support: if the issue cannot be solved at the 2nd level support, it is
escalated to the 3rd level and handled directly by the provider of the product or the problem
management team.

Figure 15. Analysis of identified software development methodologies used in the European Institutions – 3rd level support flow



Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

4.3.1.3.Handling of major incidents

In this point we will analyse if a special plan or procedure, different from the standard, is implemented for critical issues.

Table 4-13 Analysis of identified software development methodologies used in the European Institutions – Major incidents resolution

| Special Procedure | Out of the 14 analysed projects | |
|---|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Raise the priority | 13 | 93% |
| Skip environments from non-critical procedure | 8 | 57% |
| Shutdown System | 4 | 29% |

All the projects have a special plan in order to address critical bugs or vulnerabilities, as it is shown in the previous table. 93% raise the priority to the highest in order to fix the issues as soon as possible, 29% shutdown the system in order to avoid the propagation of the vulnerability or to prevent errors caused by the bugs, and 57% of the projects skip some test environments to promote the fix to production as soon as possible and solve the incident.

4.3.1.4. User notification

This point analyses the approach used to notify the users when a new bug or vulnerability is identified or when it is necessary to conduct a new deployment that could cause service interruptions.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

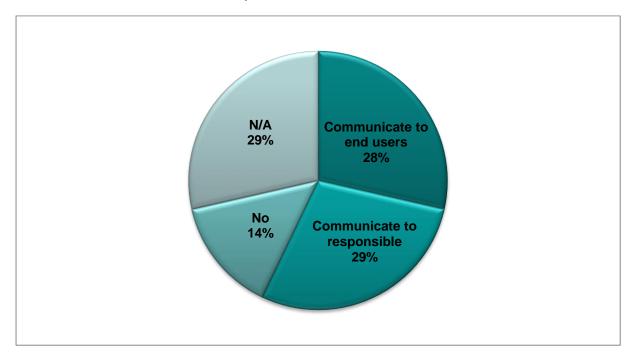


Figure 16. Analysis of identified software development methodologies used in the European Institutions – User notification

| User notification procedure | Out of the 14 analysed projects | |
|-----------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Communicate to end users | 4 | 29% |
| Communicate to responsible | 4 | 29% |
| N/A | 4 | 29% |
| No | 2 | 14% |

29% of the analysed projects notify end users about the service interruptions, 29% to the person responsible for the application on the business side, two projects use the helpdesk to disseminate the communication and 14% conduct the deployment or patch installation without notification. For the remaining projects, this variable does not apply because the projects are not yet deployed in production.

4.3.1.5. Channels and tools used

The analysed projects that report the service interruptions use the following tools:

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Table 4-14 Analysis of identified software development methodologies used in the European Institutions – Channels and tools for communication

| Channel and tool | Out of the 14 analysed projects | |
|------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| e-mail | 5 | 50% |
| Yammer | 4 | 40% |
| Static Page | 4 | 40% |
| Intranet | 3 | 30% |
| Helpdesk | 2 | 20% |
| Confluence | 1 | 10% |
| IRM | 1 | 10% |

50% of the analysed projects use e-mail to notify, 40% use Yammer, 40% use a static page which shows a predefined message, 30% use the intranet, 20% use the helpdesk, 10% use Confluence and another 10% use IRM channel.

4.3.1.5.1.Time to resolve a bug or vulnerability

This is a metric to measure how much time the development team spends fixing the bugs and vulnerabilities to ensure the quality and security of the application.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

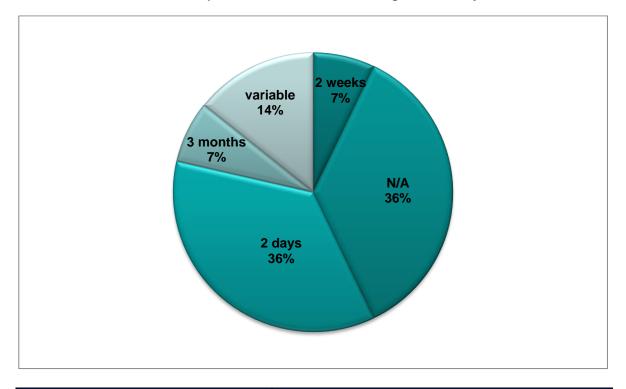


Figure 17. Analysis of identified software development methodologies used in the European Institutions – time to fix bug / vulnerability

| Period | Out of the 14 analysed projects | |
|----------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| N/A | 5 | 36% |
| 2 days | 5 | 36% |
| Variable | 2 | 14% |
| 3 months | 1 | 7% |
| 2 weeks | 1 | 7% |

This chart shows the average time to resolve a bug, for five projects is 2 days, for one project it takes 2 weeks, for another one 3 months, two projects mention that is difficult to measure because depends of the bug, and another five do not have this information.

4.3.1.6. Conclusion

Taking into consideration ITIL [2] Incident Management as a best practice, the following processes are covered by the analysed projects.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

- Incident management support: It is covered by the entry points and tools used for this purpose.
- Incident logging and categorisation: It is covered in the entry point and problem management phases.
- Immediate incident resolution by 1st level support: All the analysed projects have a least 1st level support.
- Incident resolution by 2nd level support: Some analysed projects have 2nd level support with specialized technicians.
- Handling of Major Incidents: Some analysed projects address major incidents with their own process.
- **Pro-Active User Information:** Not all end users of the analysed projects receive notification when the service is disrupted.

4.3.2. Problem Management

4.3.2.1. Identification of security updates or bugs

This point analyses how the project team identifies potential security updates or bugs in advance, as a preventive process. This point it is important since newly discovered vulnerabilities or bugs handled on time will reduce the probability of service disruptions, and thus will ensure the availability of the system and the quality of the service provided to end users.

Table 4-15 Analysis of identified software development methodologies used in the European Institutions – Identification of security updates or bugs

| Procedure | Out of the 14 analysed projects | |
|--|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Regression Test | 13 | 93% |
| Security information updated from providers | 5 | 36% |
| Security information updated from external resources | 5 | 36% |

Most of the analysed projects identify deployed vulnerabilities and bugs using regression tests, but this procedure doesn't ensure that other bugs or vulnerabilities different from written tests are addressed. 36% of the projects rely on the recommendations from the provider of the baseline application (Piwik,

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

Drupal, EMC, Alfresco) and the remaining 36% of the projects update the security information from external resources specialised in security.

4.3.2.2. Problem resolution plan

In this point we analyse how the project team responds to a bug or vulnerability that has been reported using the channels analysed in section 4.3.1.5

| Procedure | Out of the 14 analysed projects | |
|------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| Standard Release | 11 | 79% |
| Special Process | 5 | 36% |

 Table 4-16 Analysis of identified software development methodologies used in the European Institutions – Problem resolution plan

Most of the projects respond to problem resolution with a standard release procedure. The bug or vulnerability is logged as a task; the development team fixes the issue and uses the normal procedure, through the defined deployment pipeline, in order to promote the version between environments. However, three projects have a special plan to fix bugs or vulnerabilities and two projects use a combination of these two, standard release and special process, depending on the criticality of the issue.

4.3.2.3. Tools and resources used

- Selenium
- JUnit.
- OWASP.
- European Parliament Standards and Methods recommendations.
- Digit recommendations
- Provider recommendations (Drupal, Alfresco, Piwik)

4.3.2.4. Conclusion

Taking into consideration ITIL [2] Problem Management as a best practice, the following processes are covered by the analysed projects:

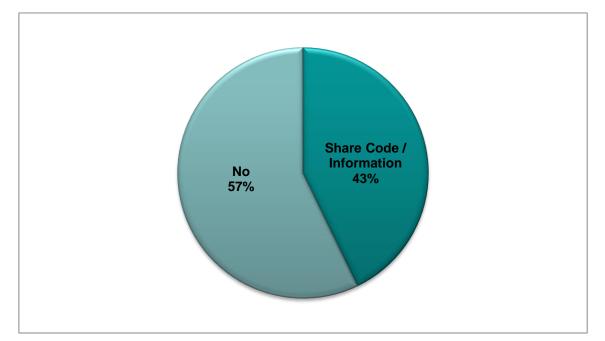
Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

- **Proactive Problem Identification:** Some projects update the security information from external resources to solve them as soon as possible. Regression tests are also used as they can provide bug identification in advance.
- Problem Categorisation and Prioritisation: It is covered within the entry point and development phases
- Problem Diagnosis and Resolution: It is covered by the IT team.

4.4. How European Institutions contribute to FOSS Communities

In this point we will consider how the analysed projects contribute to the FOSS Communities in terms of sharing code or information.





| Foss communities contribution | Out of the 14 analysed projects | |
|-------------------------------|---------------------------------|----------------------------|
| | Number of projects | Percentage of the projects |
| No | 8 | 57% |
| Share code / information | 6 | 43% |

The chart shows that 43% of the analysed projects share code or information with the FOSS Communities. Among these contributions, the most important ones are:

Document elaborated in the specific context of the EU – FOSSA project. Reuse or reproduction authorised without prejudice to the Commission's or the authors' rights. Page 107 of 110

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

- **Code sharing**: For new or updated functionalities which have been developed within the European Institutions projects, they share the code with the corresponding FOSS communities.
- Information Sharing: Detailed information about procedures, methodologies and architectures is provided to the FOSS communities.
- JoinUp: 29% of the projects are published in JoinUp [4],

This analysis shows that the majority of the contributions to FOSS communities are on a personal level as the developers do not associate themselves with the European Institutions. Two projects contribute with the FOSS communities sharing the entire application to the community licensed as EUPL. The role used in this contribution is institutionally based and it is sponsored by the corresponding DG.

One project is willing to become a sponsor for the FOSS community that develops the product that they use, PIWIK.

4.5.Relevant opinions and advices from interviewees

During the interview process some opened questions were addressed to the interviewers. The most relevant opinions are:

- 1. PM² doesn't define properly the security role.
- 2. Security and PM² should be linked
- 3. Security is coming at the end of the SDLC and it should be at the beginning.
- 4. Improve the security in the European Commission.
- 5. System security experts should report directly to the development team so security is considered from the early phases.
- 6. The European Commission should open up to Free and open source.
- 7. Create a list of supported Free and open source products.
- 8. The European Institutions should support European innovation using software companies or European FOSS communities.
- 9. The European Commission experts should provide a Free and open source replacement list for private applications.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

5. Bibliographic references

- [1] IBM, "IBM Rational," [Online]. Available: http://www.ibm.com/software/rational.
- [2] OWASP, "OWASP," Free and open software security community, [Online]. Available: https://www.owasp.org.
- [3] ITIL, "Itil Books," [Online]. Available: http://www.itil.org.uk/.
- [4] E. Comission, "joinup," [Online]. Available: https://joinup.ec.europa.eu/.

Deliverable 3: Analysis of Software Development Methodologies Used in the European Institutions

6.Annexes

6.1.Questionnaire for the interview



6.2. Executive summary

