



FP7-ICT-2011-8

MARKOS

The MARKet for Open Source

An Intelligent Virtual Open Source Marketplace



WP6 - Front-end and Validation

D6.2.1a – MARKOS evaluation: requirements, metrics, methodology and scenarios

Due date: 30.09.2013r.

Delivery Date: 11.10.2013

Author: Alicja Ciemniowska (PSNC)

Partners contributed: PSNC, ATOS, T6, UNISANNIO

Dissemination level: PU

Nature of the Deliverable: Report

Internal Reviewers: Massimiliano Di Penta (UNISANNIO), Jesus Gorrongoitia (ATOS),
Francesco Torelli (ENG)

VERSIONING		
VERSION	DATE	NAME, ORGANIZATION
0.1	17.07.2013	ALICJA CIEMNIEWSKA (PSNC)
0.2	04.09.2013	ALICJA CIEMNIEWSKA (PSNC), ILKNUR CHULANI, JAMES AHTES (ATOS), ANTONELLA PASSANI, LUCA SIMEON, (T6), MASSIMILIANO DI PENTA (UNISANNIO)
0.4	26.09.2013	ALICJA CIEMNIEWSKA (PSNC)
0.5	04.10.2013	ALICJA CIEMNIEWSKA (PSNC), LUCA SIMEON, ILARIA LENER (T6), MASSIMILIANO DI PENTA (UNISANNIO)
0.9	07.10.2013	ALICJA CIEMNIEWSKA, PAWEŁ KĘDZIORA (PSNC), JAMES AHTES (ATOS)
1.0	11.10.2013	ALICJA CIEMNIEWSKA (PSNC), JESUS GORRONGOITIA (ATOS), MASSIMILIANO DI PENTA (UNISANNIO), FRANCESCO TORELLI (ENG)

Executive summary

This report describes the evaluation plan and the set of validation actions that will be implemented during the project life cycle in order to assure the challenging solutions MARKOS has proposed really satisfy the needs and expectations of possibly widest communities of developers.

Software evaluation occurs since creating the first lines of code until the completion of the product in test. Validation itself refers to the evaluation of the prototype from users' perspective, and aims at answering the question: "Does the MARKOS system meets users' needs?".

In order to assure consensus and uptake of the evaluation results, objective common criteria and validation approach needs to be clearly defined up-front and agreed by different stakeholders involved in the assessment. This document provides the overall plan of the validation.

The plan starts from the definition of the validation objectives based on the scientific and technical goals of the project and different categories of user requirements. Subsequently, on the basis of more specific objectives the metrics are defined and different validation activities are identified to involve users and collect their feedback and ideas for possible system amendments. Finally, the plan also includes the agreed mechanism for reporting validation results to the project development team and specifying the plans for implementing the findings.

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 European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

© Copyright in this document remains vested with the MARKOS Partners

D6.2.1a – MARKOS evaluation: requirements, metrics, methodology and scenarios

Table of Contents

LIST OF FIGURES	5
LIST OF TABLES	5
1. INTRODUCTION.....	6
2. VALIDATION BACKGROUND AND OVERVIEW	7
2.1. OBJECTIVES OF MARKOS	7
2.2. VALIDATION OBJECTIVES, REQUIREMENTS AND METRICS.....	7
2.3. VALIDATION TARGET GROUPS	10
2.4. VALIDATION ENVIRONMENTS.....	11
2.5. VALIDATION PHASES	12
3. BASELINE SCENARIOS	13
4. LAB/CONTROLLED ASSESSMENT ACTIVITIES	15
4.1. PILOT ASSESSMENT	15
4.2. USABILITY WORKSHOP	16
4.3. COMPARATIVE EXPERIMENT WORKSHOP	18
5. FIELD ASSESSMENT ACTIVITIES.....	20
5.1. REQUIRED DISSEMINATION ACTIVITIES	20
5.2. FIELD VALIDATION TOOLS	21
5.2.1. <i>Online free feedback form</i>	21
5.2.2. <i>Questionnaire</i>	22
5.2.3. <i>In depth interviews</i>	24
5.2.4. <i>Prototype website traffic analysis</i>	25
6. VALIDATION RESULTS ANALYSIS AND REPORTING	28
6.1. FEEDBACK AND METRICS ANALYSIS	28
6.2. BUG/ISSUE REPORTING	29
7. CONCLUSIONS	30
8. REFERENCES.....	31
9. APPENDIX 1: QUESTIONNAIRE.....	32
9.1. SHORT “VANILLA” VERSION OF THE QUESTIONNAIRE.....	32
9.2. ADDITIONAL BUNCH OF SPECIFIC QUESTIONS FOR AN INTERVIEW OR TAILORED SURVEY	33

LIST OF FIGURES

Figure 1: Validation phases and environments	12
Figure 2: External feedback from validation will help shape Year 2 development and post-project sustainability	28

LIST OF TABLES

Table 1: Specific objectives of validation	10
Table 2: Template for finding description.....	29

1. INTRODUCTION

Deliverable objectives

This document describes the validation plan for the MARKOS project, a common approach to ensure that the challenging solutions MARKOS has proposed really satisfy the needs and expectations of possibly widest communities of developers.

This deliverable describes the high level validation plan and the set of validation activities that will be conducted during the project life cycle in order to assure the achievement of project goals. The second version of the document at a later stage of the project will be released where a detailed plan for the final validation will be presented.

Validation refers to the evaluation of the prototype from users' perspective, and aims at answering the question: "Does the MARKOS system meets users' needs?".

To properly plan the evaluation and in order to assure consensus and uptake of the evaluation results, objective common criteria and approach need to be clearly defined up-front and agreed by stakeholders involved in the assessment. The purpose of this document is to define in details the objectives, needed metrics and methodology to ponder the fulfilment by the MARKOS system.

Structure of document

The document is divided into seven main chapters, including this introduction.

Section 2 provides an overview description of MARKOS challenges, validation objectives, target user groups, environments and phases.

Section 3 describes the baseline scenarios for validation.

Section 4 provides more detailed description of the validation conducted in the environment controlled by MARKOS Partner organisations.

Section 5 describes the plan for the field environment assessment involving the wider context of developers from open source communities.

Section 6 outlines the plan for analysing validation results and communicating them to the development team.

Section 7 concludes the document.

2. VALIDATION BACKGROUND AND OVERVIEW

2.1. Objectives of MARKOS

The MARKOS project aims to facilitate the software developer's discovery and analysis of open-source software suitable for his/her technical and legal needs.

To this end, MARKOS will realize the prototype of a service and an interactive application providing an integrated view on the Open Source projects available on the web, focusing on functional, structural and licensing aspects of software code.

With respect to the offered capabilities MARKOS aims to support users as follows:

- support search of specific components using expressive enquire mechanisms based on software dependencies other than usual features such as language, operating system etc.
- facilitate the understanding and comprehension of the software from a technical point of view by a more user friendly navigation and browsing of the code
- allow a more efficient and accurate analysis of licence compatibility and to provide well founded legal argumentations
- facilitate the collaboration between different projects (upstream/downstream ticketing)

MARKOS wants to offer developers and analysts a solution for choosing the Open Source artifacts that suit their needs, allowing them to learn how to integrate or extend such artifacts and more in general to foster easy adoption of Open Source software.

To fulfil these challenges the MARKOS project intends to realize the prototype (MARKOS system) of an automatic service providing an integrated view on Open Source software available in the web, focusing on functional, structural and licensing aspects of the software code released by the projects.

The overall technical objectives translated into functional requirements are described in D1.1.1a [1]. The specification includes a set of user requirements organized in themes, epics, features and user stories. Note that during the first year of the project only part of the functionality will be implemented and thus available for validation.

2.2. Validation objectives, requirements and metrics

Software verification & validation is a part of development life cycle since writing first lines of code until the completion of the product in test. Specifically, validation aims at answering the question “Does the MARKOS system meets users’ needs?”. As a consequence, in order to answer such a question, the validation needs to involve real users. Instead, verification aims at checking whether developers have “built the system right” i.e., whether the system conforms to the requirement. Both aspects are necessary as there is no sense in building a technically correct system that is not useful for the intended users. In MARKOS project verification is an integral part of the development process, and therefore not in the scope of this task.

The overall goal of the validation task is to conduct a fit-for-purpose evaluation that will assess the MARKOS system in real life scenarios with involvement of FLOSS developers to assess the MARKOS scientific and technical challenges satisfy the needs and expectations of users. It will focus on validation with users checking if the system is appropriate for the purpose it has been designed and meets the business and operational needs; as a consequence, the system can be used in real-world scenarios, e.g. by OSS communities.

The results of the project will be evaluated in the context of real life situations in which MARKOS services will be exposed to the community of users. For this purpose MARKOS will provide an online prototype service where users can play with the tool in their everyday working activities.

The typical categories of requirements related to software that are considered in MARKOS validation process to specify detailed objectives of user acceptance validation are as follows:

- Functionality
- Information
- Usability
- Performance
- Benefit

These subcategories are described in more detail below. For each objective, a description in terms of research questions and metrics that will be collected during validation.

Specific objectives to validate with end users:

O.1 Functionality

To assess if the capabilities of MARKOS prototype meet users' needs

Do the capabilities of MARKOS prototype meet users' needs or they have to be modified or deployed in some other manner? What do people like/dislike about the MARKOS searching, browsing and license analysis functionality? What additional pieces of functionality have to be prioritized (that is in the scope of the project)?

Metrics:

- *Users subjective level of usefulness w.r.t. to capabilities*
- *Number of reported bugs, new features and modifications*

<p>O.2 Information</p> <p>To assess if the information provided by MARKOS system is the kind the users look for</p>	<p>How satisfied are users with the content provided, usefulness and adequacy of information? Do they miss any information?</p> <p>Metrics:</p> <ul style="list-style-type: none"> - <i>Users subjective level of satisfaction and perceived usefulness of information</i> - <i>Accuracy and completeness of the results provided by MARKOS when users perform queries</i>
<p>O.3. Usability</p> <p>To assess the prototype meets needs with respect to design and navigation</p>	<p>Is the prototype easy to learn and use, simple to understand? What users like/dislike about the MARKOS design and navigation?</p> <p>Metrics:</p> <ul style="list-style-type: none"> - <i>Usability problems and suggestions</i> - <i>Use errors</i> - <i>Task Level Satisfaction</i> - <i>Time required to perform a task</i>
<p>O.4 Performance</p> <p>To ascertain the prototype meets basic performance requirements</p>	<p>How a system performs in terms of responsiveness and stability under a particular workload?</p> <p>Metrics:</p> <ul style="list-style-type: none"> - <i>Users subjective level of satisfaction w.r.t. to response time, performance, reliability</i> - <i>Number of simultaneous users</i>
<p>O.5.Benefit the MARKOS will provide</p> <p>To assess the perceived value of MARKOS system and the benefits it provides to users)</p>	<p>Is the system appropriate for the purpose it is designed? Does MARKOS provide added-value to the current workflow to discover, analyse and integrate OSS projects? What the benefits users see from using the MARKOS system?</p> <p>Metrics:</p> <ul style="list-style-type: none"> - <i>User overall satisfaction</i> - <i>Users willingness to pay for the service</i> - <i>Time/cost reduction</i> - <i>Improvement of working routine</i> - <i>Number of returning visitors</i> - <i>Impact of MARKOS (on output quality, on human capital and company/professional reputation, on software development value chain and on the relationship with other companies/professionals</i>

Table 1: Specific objectives of validation**2.3. Validation target groups**

The main users of MARKOS will be software developers who adopt Open Source software or who are directly involved in Open Source development communities.

In order to assure the MARKOS innovative solutions really satisfies the needs and expectations of the widest communities of developers MARKOS assessment and evaluation needs to involve as much users as possible and with different background. MARKOS outcomes are targeted not only to existing OSS communities, but also to the wider software and services IT landscape at large. Therefore, the field validation activities will be addressed to users using and integrating open source software in both industrial and scientific domains in order to get the perspective of users from different open source communities. The industrial scenarios should include commercial Independent Software Vendors of various size, from large enterprises to self-employed businesses, IT services, system integration, consultancy, SMEs etc.

The following kinds of user have been identified during the requirements specification:

- Software Consumer - interested, at least, in information about software and projects.
- Software Architect - interested also to any information useful to design or understand the architecture of a software system.
- Software Integrator - interested to any information useful to integrate different software component.
- Software Developer - interested to any information useful to develop a new software component or to evolve an existing one.
- Project Manager - manager of one or more FLOSS software projects indexed by MARKOS.
- Administrator - taking care about administration of the MARKOS system.
- Copyright Lawyer - A professional lawyer or attorney with expertise in copyright law and, in particular, Open Source licence compatibility issues.

Basically the above listed types of users can be divided in groups with two different backgrounds: **technical IT or legal**.

As mentioned above the main users of MARKOS will be software developers who adopt or are directly involved in Open Source development communities. Therefore most validation activities will targeted to the group of technical IT people.

It should be also mentioned that, as described in more detail in Section 3, during the first year only part of the functionality will be implemented and thus available for validation. In particular features addressed to the copyright lawyers will not be available in the first prototype. The license checking and analysis features included in the first prototype are

addressed to developers that tries to deal with open source licenses without professional legal advice. For this reason the validation activities performed based on using the first prototype will be oriented mainly to the technical IT people developing software. It is assumed they have (or they want to get) a basic understanding of copyright law. The copyright lawyers will be involved in prototype validation in later stage of the project or through more flexible tools, such as the in-depth interviews.

The involvement of different types and groups of users will be done by targeting the dissemination activities to those groups.

2.4. Validation environments

Specifically, the evaluation and assessment in MARKOS will be performed in two different environments:

- *lab/controlled* environment assessment where there will be pilot assessment in the MARKOS partner environment and controlled workshops
- *field* environment assessment implying a real world context assessment with end users, with the involvement of a large set of evaluators invited to test MARKOS tools.

The lab/controlled environment will be used to perform pilot controlled assessment or perform controlled workshops under certain lab conditions. Firstly, the controlled pilot evaluation will make sure everything is running fine and is ready to be exposed and demonstrated to the end users and the mechanism/tools to collect feedback. It will be used to check technical aspects of the prototype, such as response time, compliance of the results, and usability of the functionality offered. Secondly, the controlled environment will allow for more in-depth validation activities on usability aspects. Last, but not least, the controlled comparative experiment will allow to perform an evaluation in which MARKOS will be directly compared with an alternative scenario in which developers do not use MARKOS, but rely on tools they already use in their activities, e.g. Internet Search engines and specialized forums such as StackOverflow.

Field trials are crucial for the complete validation. Making potential end users play with the prototype can help to identify some lack in the specifications or something that the developers have not taken into account. The goal of the field assessment is to evaluate that it's challenging solutions really satisfies the needs and expectations of the widest communities of developers. By the field testing exercise the MARKOS system will be evaluated in the context of real life situations in which we will expose its services to community of developers both from industries and Open Source communities, including Open Source communities of developers such as BerliOS and SourceForge.net.

2.5. Validation phases

Usually in projects and companies, the validation if the system meets the users' needs is done late in the project, because development team are afraid of showing not mature system and expect a lot of criticism. However, validation is a continuous process that should start early, even before writing first line of code, from the requirement elicitation and specification phase. Moreover, the validation needs to be done throughout the development life cycle of the system, because the system changes so as the user needs.

In the overall MARKOS development life cycle we distinguish two validation phases. The outcome of each phase is a report summarizing the results from the planned activities and feedback collected from users.

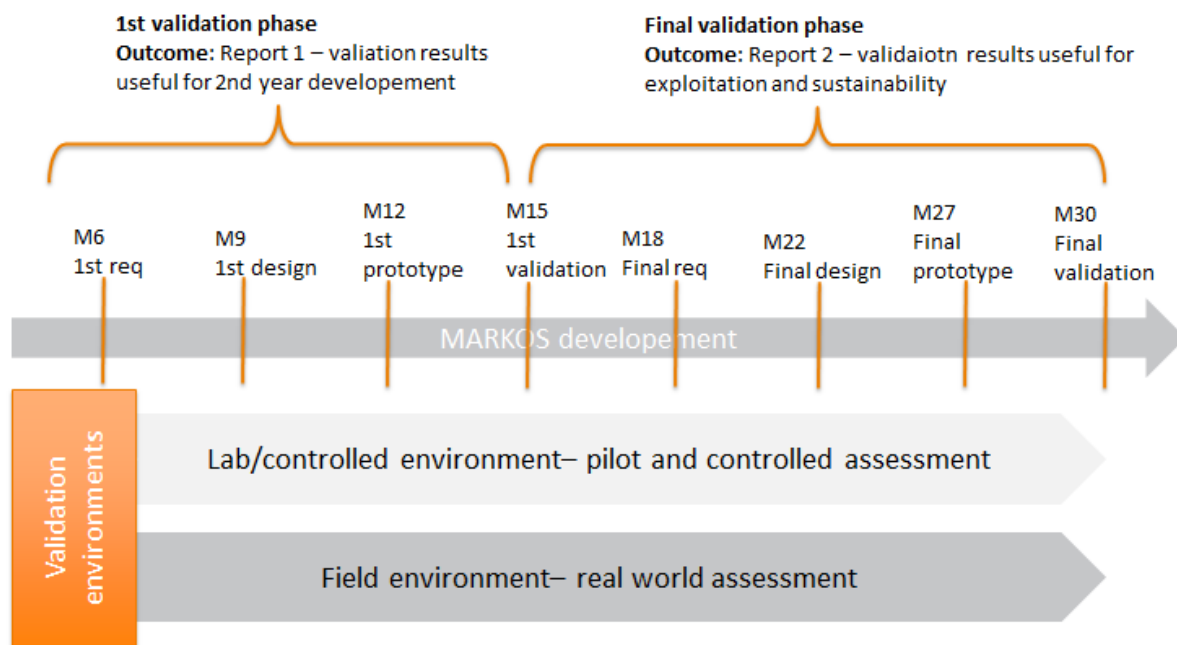


Figure 1: Validation phases and environments

The first validation report will be released in M15, three months after the release of the first MARKOS prototype. This report will provide information and findings useful for the further development. On the basis of the 1st validation results, the requirements will be updated and, subsequently, a new set will be selected to be (re)implemented in the next development cycle.

The second and final deliverable is scheduled for the end of the project. Therefore the information in the report will be useful for the exploitation and sustainability activities.

Despite the two reports summarizing the outcomes, the MARKOS validation will be conducted in a more iterative process. The different validation activities will be conducted throughout the software development life cycle, based on the results available. The findings will be communicated to the development teams on regular basis.

Some initial validation activities have been already organized in the first year of the project, survey on usage scenarios, described in [1], and a focus group interview with end users. As

the first public release of the prototype will be available at the end of the first year, the subject to validation before the release were the MARKOS system requirements.

Starting from second year, the next validation activities will be conducted mainly on the basis of the software prototype.

3. BASELINE SCENARIOS

The overall technical and scientific objectives of MARKOS have been translated into functional requirements expressed in D1.1.1a [1] in a set of scenarios and user requirements organized in themes, epics, features and user stories.

During the first year only part of the functionality will be implemented and thus available for validation. Below there is presented the usage scenario, derived from the original scenarios published in [1], but cut down to the functionality implemented during the first year of the project, which are being the subject to first validation phase activities i.e.

The scenario will be used as a basis for defining more concrete tasks in the lab-controlled environment as well as provide some help to the real users in the field assessment.

Scenario description

Antonio is a software developer working on his first Java swing application. In this context, he needs to implement a file hierarchy viewer (i.e., a tree view), showing all files in the system on which its application is running. From this view, it should be possible to open all textual files by performing a double click on the leaf representing it in the showed tree. He knows that the swing `JTree` class can be used to implement such a feature and, after few attempts, he successfully implements the hierarchy view. The next challenge is to implement a listener able to open the textual file eventually double-clicked by the user. Looking at the swing documentation Antonio discovers that there is an API, called **`javax.swing.event`** that contains all the interfaces that could be implemented to realize a listener notified whenever a selection in a `JTree` changes. However, Antonio has no idea on how to implement this API.

Thus, Antonio accesses to the MARKOS service, writes the name of the API he is interested in (i.e. **`javax.swing.event`**), selects exact match and **API** as constraints.

He gets one result for which he wants to visualize the details. He notices that differently from other web tools, the entities of the API implementation are well distinct from the entities of the API and there is less redundancy in the results.

Antonio selects one of the results and browses the project releasing it to analyze dependencies among the classes and the inheritance relationships. He notes that it is possible to see and browse the part of relationship between software entities: something that is not possible with other web tools. Once shown the details he sees the ***Find all API implementations*** option. In one click he found several packages from different projects implementing the API he looked for.

Markos shows the list of results:

- `org.apache.cayenne.modeler.dialog.objentity`

- org.apache.cayenne.modeler.editor
- org.apache.cayenne.modeler.editor.dbentity
- org.apache.cayenne.modeler.undo
- org.apache.cayenne.modeler.util
- org.objectstyle.cayenne.dataview.dvmodeler
- org.objectstyle.cayenne.modeler.editor.dbentity
- org.objectstyle.cayenne.modeler.util
- org.apache.cayenne.dataview.dvmodeler
- org.apache.jmeter.visualizers
- more ...

Antonio browses the results to understand how people implements the API. He finds that *viva.ui.ot.gui* perfectly fits his needs. However, the License Checker points out this implementation has some legal issues. To this aim, Antonio opens the License Analyzer to see the legal argumentation and understand the causes of such legal issues, and realizes that this is because *viva.ui.ot.gui* contains code released under GPL.

Since Antonio cannot use reciprocal licenses, he tries to browse another result. He notices that the result uses an Apache license and there is no legal issues. He found an implementation that suits his needs.

4. LAB/CONTROLLED ASSESSMENT ACTIVITIES

In the lab/controlled assessment the MARKOS prototype will be used and tested under controlled conditions by real end users from MARKOS partners organisations and their connected communities.

The controlled environment will be used to prepare for the field assessment, by conducting a pilot assessment, as well as will allow to gather more in-depth feedback form users on the specific aspects of MARKOS prototype thanks to organization of usability and usefulness workshops with end users.

4.1. Pilot assessment

Objectives covered	O.1 – O.4
Target user groups	Users with technical IT background
Number of users	14 (two from each partner organisation depending on size)
Timeframe	M6-M14

Prerequisites:

The online prototype demo and setup of all mechanisms for collecting user feedback prepared for field assessment.

Environment:

The pilot assessment will use the field environment assessment tools for collecting the feedback. The pilot assessment will use the same instance of MARKOS system that will be used by the external users.

Metrics:

All metrics collected during the field assessment with the respective feedback collection tools.

- Critical issues or fixes that need to be addressed by developers immediately in before field testing.
- Issues and improvements needed to feedback mechanisms.

Description:

The pilot assessment will be performed to identify all possible issues/bugs before making available the prototype to the end users external to the project. The focus of this testing is to act as real users and to carry out the tasks that a typical user might perform. The pilot assessment will be carried out in the same environment that will be available to external users,

but the testers will be the MARKOS partners organizations. The aim is to carry out the tasks that a typical user might perform.

The pilot assessment will be a pre-check of the reliability, response time, compliance of the results, and usability of the functionality offered before publishing the prototype to the wider community.

This test should be performed by at least two testers from each partner organisation.

The users will get a link and access to the online published prototype and should play with it. They are asked to provide feedback by both fulfilling the online questionnaire, described in section 5.2.2, or by sending free comments through the online feedback form, described in section 5.2.1. High priority issues or bugs should be reported the development team using the Manits system.

Expected outcome

Critical issues or fixes that need to be addressed by developers immediately in before field testing. Issues and improvements needed to feedback mechanisms.

4.2. Usability workshop

Objectives covered	O.3
Target user groups	User with technical IT (and legal background in the second stage)
Number of users	5 in each of the two workshops planned
Timing	M16-M20 and second test after 5 months

Prerequisites:

- Definition of the requirements to be tested, based on functionalities available after M18
- An interactive prototype of the system with access to the functionalities to be tested has to be available
- Suitable protocols for the tests defined
- A set of users and suitable locations have been selected

Environment:

Ideally, a lab specifically equipped for user testing would be needed. If this were not available, the environment will be set up in a quiet room, equipped with two computers (one for the facilitator and one for the user) and a video camera.

Metrics:

The metrics we will consider are:

- Usability Problems
- Use Errors
- Task Level Satisfaction

Description:

The usability tests will be targeted to assess the following three components of the user experience: learnability, efficiency and satisfaction.

The tests will involve real users (10, across 2 different tests). The schedule of this test is linked to the development schedule.

We propose two different protocols for the two tests:

- Test 1: free navigation with thinking aloud protocol
- Test 2: task analysis with thinking aloud protocol

In the first phase (early prototype) a first usability test will be conducted using free navigation with thinking aloud protocol (in a lab setting, the user and a facilitator sit together in front of the UI and the user is let free to navigate it while she comments on what she is experiencing. The facilitator's role is to let the comments emerge in order to assess some qualities of the user interface).

At a second stage, when we have a more structured prototype of MARKOS, a second usability test will be conducted using a task analysis method, where - always in a lab setting - users are assigned specific tasks and - always using the thinking aloud protocol - they verbalize their thoughts and feelings while performing these tasks. The task will be defined at a later stage, based on functionalities available after M18.

The activities related to the preparation and conduction of tests are:

- Definition of suitable protocols for the tests
- Selection of users
- Set-up of suitable locations
- Identification of one or maximum two facilitators
- Conduction of the tests (in both tests 5 users will be invited and the tests will last for about 5 hours – with 1-hour slot for each user)

- Analysis and production of the final outcome, a usability report that also addresses eventual shortcomings of the user interface proposing some solutions

Expected outcome

Usability report with indications on how to improve (eventual) critical areas.

4.3. Comparative experiment workshop

Objectives covered	O.1-O.5
Target user groups	Users with technical IT background
Number of users	Between 12 and 30 or even more
Timing	Phase 2, M18-M20

Prerequisites:

The MARKOS platform should be usable enough, which means that (i) we have addressed issues related to the user interface (5.6), the response time is not very long (otherwise developers would just abandon MARKOS , and (iii) the MARKOS repository has been populated with a large set of projects, or at least with projects that properly cover possible components that developers would use during their task.

Environment:

The tasks developers have to perform should be some small implementation/maintenance tasks in which it will be necessary to find implementation of a given API.

A lab equipped with several desks and chairs where users could perform these tasks in comfortable conditions is needed. It is assumed that the users will use their own laptops.

Metrics:

- Time required to perform the task
- Completeness and correctness of the task. The idea is that, given the set of sub-tasks that compose a task, we would compute the percentage of sub-tasks that have been correctly completed.
- Responses from a survey questionnaire asking developers (i) how much time did they spend by using MARKOS during their development tasks (ii) to what extent did they found it useful; (iii) a comparative evaluation with respect to performing the same tasks without MARKOS e.g. searching for API implementations on the Web.

Description:

In summary, the idea is to perform an experimental evaluation in which we combine a quantitative analysis (time and completeness/correctness) with a qualitative one based on developers answers to a post-study questionnaire and based on focus-group interviews performed after the experiment.

We will ask the study participants to perform, each one, two different development and maintenance tasks, one using MARKOS and one without using MARKOS. Overall, we would need between 12 and 30 participants, performing tasks in two laboratory sessions of 3-4 hours each. Given the two tasks, T1 and T2, and the two treatments, MARKOS (M) and no-MARKOS (NM), the study design is the following:

Session	Group A	Group B	Group C	Group D
1	T1-M	T1-NM	T2-M	T2-NM
2	T2-NM	T2-M	T21NM	T-M

The two tasks should be related to development activities similar to those described in the MARKOS motivating scenarios. For example, these could be development tasks in which the developers are forced to use an interface, and have to find available implementations of such an interface. Moreover, the tasks could restrict the kind of licenses that can be used. Last, but not least, in order to make a comparison possible, the two tasks should have a similar complexity and, to avoid learning effect, should be independent from each other.

Expected outcome

Report summarizing the workshop findings.

5. FIELD ASSESSMENT ACTIVITIES

Thanks to the field testing exercises the MARKOS system will be evaluated in the context of real life situations and scenarios in which we will expose its services to community of developers both from industries and Open Source communities.

The prototype will be available online linked from the project website <http://www.markosproject.eu/> so users could try to use the system by themselves.

A campaign to advertise about the release and asking for volunteers interested in trying and assessing MARKOS system will be launched, as described in section 4.1. The involvement of the developers in MARKOS evaluation should be easier thanks to the involvement in the MARKOS project of Geeknet and Fraunhofer which respectively owns Sourceforge.net and Berlios, and then involve their reference communities in this field assessment.

To understand their needs we will use different field validation tools, such as:

- Free feedback form – available online to collect any kind of feedback from users
- Short questionnaire - publish online (or printed) to collect feedback in a more structured way
- In-depth interviews - done in face to face environments but also online or by phone
- Web analytics and tracking code at the pilot site to monitor and analyse how the users use the MARKOS prototype.

The field validation tools are described in more detail in section 5.2.

5.1. Required dissemination activities

Within the MARKOS project we consider Dissemination and Validation strictly connected in a reciprocal exchange of synergies.

On this regard, as described in the Dissemination Plan [2], dissemination and validation will interact according to the following guidelines:

- On the one hand, the dissemination will aim at promoting the full involvement of different stakeholders in the external validation process,
- On the other hand, assuring the collection of feedbacks from external users, the external validation is expected to support the dissemination in addressing contents of forthcoming dissemination materials. Inputs from external validation will make possible the identification of added values brought by the usage of the MARKOS system, which will be part of the message of the dissemination campaign.

With regard to the dissemination activities aimed to present MARKOS to the audience in order to recruit people interested into validation and getting useful feedbacks, the following are the ones we plan to organize at the moment:

- Webpage about field trial with link to prototype from project website
- News at OSS blogs
- Press Release
- Newsletters inviting to test prototype
- MARKOS Presentations at events.

On this regard, at the moment of writing this report, the Consortium is working to assure the participation to the following events and related dissemination activities, which will be used to show case MARKOS, enlarge the community of followers and to recruit people for the validation.

In the nearest future the MARKOS project will be present at the following events:

- Open World Forum 2013, Paris: 3-5 October 2013 <http://www.openworldforum.org/en/>
- OW2'con 2013, Paris: 13-14 November 2013 <http://ow2.org/view/OW2con-2013/>
- fOSSa Conference, Lille: 20-22 November 2013 <https://site.inria.fr/fossa2013/>
- ICT 2013 Conference, Vilnius: 6-8 November 2013 <http://ec.europa.eu/digital-agenda/en/ict-2013-conference>. MARKOS will be present to the event with:
 - Networking Booth: on the 7 November, 2 p.m. (Booth n. 2) to discuss on "Future services to facilitate FLOSS development and adoption by EU research and business communities"
 - Exhibition Desk (Stand n. 4A10), to showcase first MARKOS's prototype capabilities

5.2. Field validation tools

5.2.1. Online free feedback form

Objectives covered	O.1 – O.5
Target user groups	All users of the MARKOS system
Number of users	To scale with open prototype; objective is for at least 25% of users to complete form. If objective not met, the form

	will be deployed in another manner in order to capture the feedback of a larger number of users.
Timing	Phase 1 & 2

Prerequisites:

Feedback form will be deployed and a link from the MARKOS front-end interface will grant access to the feedback form.

Environment:

No particular environment setup is required. To access the system users need a computer with Internet connection and an Internet browser.

Feedback form will be deployed online and available from the MARKOS front-end interface.

Metrics:

The form will be an open field for both objective and subjective feedback. While the more targeted questions will be part of questionnaires (see below section 5.2.2), this form will be an open forum for sending the MARKOS team feedback of the concept and experience on qualitative merits.

The metrics collected through this tool include:

- Number of feedback messages
- Reported bugs, new features and modifications
- Reported usability problems and suggestions

Description:

The form will provide an open channel between the prototype users and the MARKOS team. Feedback gathered here will be analysed and fed back into the requirements, development priorities and sustainability path.

Expected outcome

This tool will provide open feedback, and done in parallel to the more structured questionnaire exercise described below.

5.2.2. Questionnaire

Objectives covered	O.1 – O.5
Target user groups	All users of the MARKOS system

Number of users	>50 in Phase 1
Timing	At launch of available prototype at M13, as well as tailor-made versions for relevant venues.

Prerequisites:

A modular questionnaire will be made to fit all target user groups and validation objectives, in collaboration with the work package leaders that create the MARKOS features.

Environment:

No particular environment setup is required. To access the system users need a computer with Internet connection and an Internet browser.

The questionnaires will be deployed online via Google Forms. The modular design will be also be used as a “script” for the MARKOS team during interviews (described in the next section 5.2.3).

Metrics:

Contrary to the open feedback form, this questionnaire will be made to capture particular metrics on all objectives listed in Section 2. However, it should be noted that the questionnaire will not try to replicate performance-related metrics and other more quantitative feedback that will be already covered by the portal traffic analysis (see section 5.2.4).

Description:

The modular questionnaire will be a full collection of questions gathered from the MARKOS team to help feed structured feedback in the project’s development cycle. It will be in parallel to the more open, ad-hoc feedback form of the previous section.

This will then be tailor made for the opportunity at hand, taking into account focus and time available. For example, a workshop on user experience will take into account the questions most geared for that objective. A conference of legal experts will likewise only use questions focused on that target group.

In addition, a “vanilla” version, short questionnaire focused on general questions, will be deployed online on Google Forms and its link circulated by email. However, the MARKOS team recognizes the very low participation rate of such surveys, and will therefore focus its effort on the tailor made versions.

The questionnaire structure is presented in the Appendix 1.

A first attempt to user survey have been made in M6 (Spring 2013), when an initial questionnaire was prepared based on the usage scenarios. The goal was to collect the first feedback on requirements. The results have been use to prioritize the implementation tasks for the first prototype.

Expected outcome

The questionnaire will provide a targeted feedback.

5.2.3. In depth interviews

Objectives covered	O.1 – O.5
Target user groups	Users with technical IT and legal background.
Number of users	The number of persons reached depends on the number of persons reached by other activities as this is not seen as an independent activity. In any case, the minimum number of persons to be reached is between 10 and 20.
Timing	<p>The date of the event depends on the date of other validation and dissemination events. The in depth interview can be performed after the usability workshop and, in some cases, after the public presentations of MARKOS.</p> <p>The first focus group interview has been performed in M9.</p> <p>The next interviews, both individual and group, will be organized during the second year of a project. The interview will be co-organized with dissemination events, usability testing and with comparative assessment.</p>

Prerequisites:

Users need to interact with MARKOS before answering the questions.

Environment:

No particular environment setup is required. To access the system users need a computer with Internet connection and an Internet browser.

In depth interviews can be done in face-to-face environments but also online or by phone.

Metrics:

The interview is the most flexible validation mechanism that may be adopted to the situation and the profile of the interviewed users. Therefore during this activity different metrics related to all specific objectives, listed in section 2.2 can be collected.

Description:

Thanks to the flexible nature of the interview, in this activity we are interested in collecting as reach as possible feedbacks about MARKOS potential benefits.

In depth interview allow the interviewer and the respondent to develop reach interaction and to collaboratively develop a shared understanding of the topic under discussion avoiding misunderstanding and misinterpretations. In an in-depth interview the interviewers develop a set of questions and leave the answer open to the respondent without proposing alternative answers. Also, the interviewer has a certain degree of freedom in adding and changing questions following the interaction with the interviewed.

The modular questionnaire, as described in 5.2.2 and Appendix 1, will be also be used as a part of the “script” for the MARKOS team during interviews.

Some in-depth interviews have been carried out during the first year. It included a focus group session involving software developers from PSNC.

A first group interview, conducted as a focus group discussion, have been organized in the first year of the project, in Poznań PSNC. As the prototype has not yet been available, the system was presented to the users by a technical coordinator with a slide show presenting MARKOS features including the early mock-ups of the system. The goal was to understand users opinions and feelings about MARKOS system concept and to discover what users want from the system. As well as to assess how users will use the system and what aspects are important for them when searching for and analysing the software to reuse. As an outcome of that analyses MARKOS is considering to extend the functionalities regarding the management of user feedback, actually adding to the goal of improving the collaboration among projects (see section 2.1) also to improve the collaboration between projects and users. The “Questions related to user comments capabilities” reported in appendix are a way to explore this possibility. The results will be described in more detail in the next deliverable - D6.2.2a MARKOS evaluation report.

Atos will conduct an internal focus group of the prototype with 10 selected employees in the winter of 2013/14, whose roles match that of a primary target user of MARKOS: developers using the platform to discover and analyze OSS assets to leverage for their developing software. As well, to cover the legal analysis scenarios, a representative of Atos Spain’s legal department will also participate. In order to avoid bias, the members of the focus group will have no previous knowledge of the MARKOS project.

Expected outcome

The interviews will be recorded (if possible and if the users agree) and the answered elaborated in a report aggregating the answers accordingly to questions asked and emerging dimensions.

5.2.4. Prototype website traffic analysis

Objectives covered	O.1 – O.4
Target user groups	All users of the MARKOS system
Number of users	We expect to reach several dozen users with the first prototype during 2 months.

Timing	Phase 1 (M14-M15) and Phase 2
---------------	-------------------------------

Prerequisites:

The tracking to the prototype website need to be setup using the tracking code snippet added to each page.

Environment:

No particular environment setup is required at user site. To access the system users need a computer with Internet connection and an Internet browser.

The server with the MARKOS prototype service need to be configured to collect the site visits data.

Metrics:

- Number of MARKOS prototype visitors (unique, returning)
- Time spent using the MARKOS service
- Track usage of different MARKOS capabilities

Description:

Many users, although trying a software or service like MARKOS, are not eager to provide feedback, neither through a structured survey/interview nor through a simple free feedback form. From many reasons they prefer to stay silent. The only way to get some insight from such users is to track the usage of the system using a website analytics tools to analyse how the user interact with the website.

To collect such kind of feedback regarding the usage of the MARKOS prototype service we will install and use a website analytics software (such as Google Analytics or Piwik). The tool will allow to collect and monitor website traffic, visitor data and information on site visits about MARKOS prototype.

The website analytics software reports on how visitors come to your website (e.g. via search engine, a marketing campaign, a click through from another website, social media, etc.), what they do on the website (pages viewed, frequency of visits, depth of visit), how they convert (e-commerce and other goals) and general visitors information (engagement, location, settings used, returning visitors, and more).

Such information will allow to measure user satisfaction based on metrics such as the total time spent of the website, number of returning visitors and tracking capabilities used by the users.

Additionally, it will allow to measure effectiveness of the dissemination campaigns by monitoring how visitors came to the MARKOS website.

Expected outcome

The data will be collected in the web analytics tools and elaborated in a report aggregating the data.

6. VALIDATION RESULTS ANALYSIS AND REPORTING

6.1. Feedback and metrics analysis

Feedback generated by the several means and metrics described above will be analysed and pushed towards the requirements, development priorities and sustainability path of MARKOS.

This analysis will be led by the validation task (within WP6) and in strong collaboration with the MARKOS development team, documented in the deliverable series D6.2.2 at the end of each of the two validation phases.

Validation will take advantage of the cyclical development methodology of MARKOS. Naturally, the first validation phase will shape Year 2's development, while the second phase will be more oriented towards the post-project sustainability and adoption of the platform.

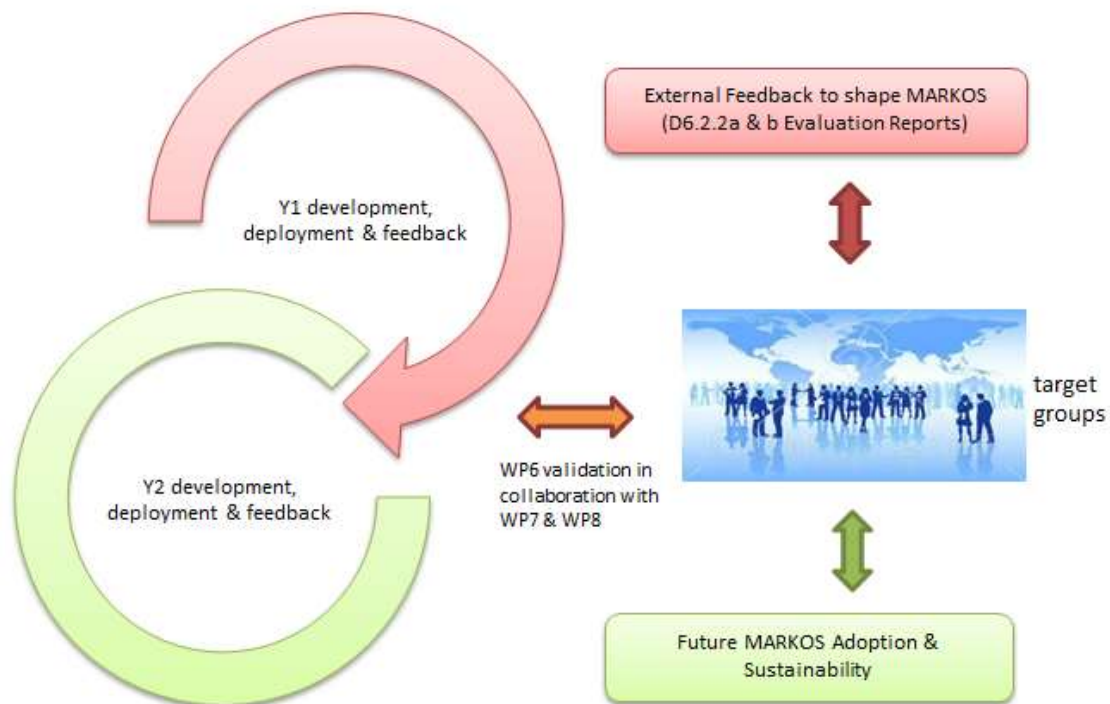


Figure 2: External feedback from validation will help shape Year 2 development and post-project sustainability

The process will be a 3-step workflow:

1. Validation results will be dissected by WP6 into specific “findings” using a standard template (see table below), and prioritized based on the correlation and repetition of particular recommendations.

2. The results will be presented to the rest of the MARKOS team, to discuss the pros & cons of adopting such recommendations, weighed against the existing development path and scope of the project. This will not be limited to just the development team, but also the sustainability path, since the validation target groups in many cases double as potential early adopters.
3. The results of the validation discussions will be specific implementation actions and timings for approved recommendations, pushing the validation activity directly towards shaping MARKOS.

Finding #No	<finding summary>		
Reasoning	<description of finding and data and comments (e.g. from survey or workshop) the finding is based on>		
Priority	<priority (low, normal, high, urgent)>		
Reported on	<date>		
Action needed by	<role in charge of implementing the finding (e.g. Scenario Leader/Component Leader... >	Timing	<planned time of implementing the finding> to be filled by assigned role
Way of implementing	<Description> <i>How the finding will be addressed? If it will be implements</i> <i>What components are involved?</i> <i>To be filled by assigned role</i>		

Table 2: Template for finding description

6.2. Bug/issue reporting

Any bug/issue detected by the validation team or reported by users will be immediately reported to the development team. For this purpose, the validation team will use of the Mantis issue tracker tool, which is used in the project to capture, discuss and finalize the user requirements and to manage their implementation.

It should be noted though that the primary mandate of WP6's validation task is not to identify bugs, but instead push external feedback towards the project's development, revolving around the validation objectives laid out in Section 2.

7. CONCLUSIONS

This document described the MARKOS evaluation approach and coordinated validation plan. The evaluation plan is a document that will serve the validation team and everybody involved in the operational evaluation phase of the project as the primary point of reference.

In this deliverable we provided an overall evaluation plan and an outline of the validation activities in the first and final validation phase.

The document described the validation objectives with respect to the expected project results and different categories of users requirements and the way the evaluation and validation activities will be organised and carried out in order to address these objectives.

The project has therefore adapted an evaluation approach that will be performed in two different environments:

- A lab/controlled environment assessment where there will be pilot assessment in the MARKOS partner environment
- A field environment assessment implying a real world context assessment with end users, with the involvement of a large set of evaluators invited to test MARKOS tools.

A second version of the validation plan will be released close to the end of the project, when the MARKOS prototype will be in a more mature stage to plan the final validation activities.

8. REFERENCES

- [1] D1.1.1a ‘Definition of storing, browsing and querying requirements’, MARKOS project, 2013
- [2] D7.1.1 ‘Dissemination Plan’, MARKOS project, 2013

9. APPENDIX 1: QUESTIONNAIRE

Below, in section 1, a draft set of questions is collected to be used in a short questionnaire. The short questionnaire is focused on general questions related to the MARKOS system and its benefits to users. It will be published online and available at the MARKOS prototype service website.

Section 2 below contains a poll of additional, more specific questions that can be used for profiled in-depth interviews or tailored questionnaires. The list is a full collection of questions gathered from the MARKOS team to help feed structured feedback in the project's development cycle.

This is a “living list” that we will keep updating.

9.1. Short “vanilla” version of the questionnaire

Date of fulfilling the survey: <open>

Name: <open>

Email address: <open>

Please tell us about yourself:

- Select your role in a software development process: <multiple choice: Policy maker / Project Manager / Software Analyst / Software Architect / Software Developer / Software Integrator / Data Integrator / Software Consumer / Legal expert / Administrator / Other (please provide a name)>
- What other platforms do you use to search for and analyze open source (Check all that apply)? *koders.com / sf.net / Berlios / ow2 / Join up / Ohloh / Ohloh Code Search / Google / Google Code Search / GrepCode / Antelink / Palamida / OSCAD / devrates.com / StackOverflow / other*

What aspects are important for you when you search for and analyse an open source code to reuse? Please rate the perceived usefulness of each of the following MARKOS attributes and capabilities: <scale from 1 “Highly useful”, 2 “Useful”, 3 “Somehow useful”, 4 “Useless” to 5 “Highly useless”>

- General searching
- Filtering by entity type (Class, Interface, Library, API, ...), programming language, license and its category, tags...
- Advanced searching
- Software structure browsing and navigation
- Abstract description of the source code independent from a specific programming language
- Finding all API implementations (available in API view)
- Software dependency and provenance analysis
- Finding compatible licenses
- In-depth analysis of licenses compatibility issues
- Aggregation of information from different sources

- System design and navigation
- Overall system reliability and performance
- Other users' comments

Please give us your opinion on the following statements: <scale from 1 “Strongly agree” 2 “Agree”, 3 “Neither agree nor disagree”, 4 “Disagree”, 5 “Strongly disagree”>

- MARKOS can provide added-value to the current workflow to discover, analyse and integrate OSS.
- MARKOS can support the searching of software components by its expressive enquire mechanisms.
- MARKOS can facilitate the understanding of software components, their structure, interfaces and dependencies.
- MARKOS allows a more efficient and accurate analysis of license than the currently available sites.
- MARKOS can facilitate the collaboration between different projects
- MARKOS can provide enough information to make decision whether to use a particular open source software component.

Open questions:

1. Please explain in which way the MARKOS capabilities can impact your working routine and what benefit it could bring <open>
2. What functionality is missing that would be useful in software searching and technical and legal analysis? <open>
3. What design, navigation or content elements need to be improved? <open>
4. Did you experience any problem while using the MARKOS prototype? Yes (Please explain the issue) /No : <open>
5. What other improvements you would like to see made to the prototype? <open>

9.2. Additional bunch of specific questions for an interview or tailored survey

Questions related to searching capabilities

- Would you like to have the possibility to submit SPARQL queries on software structure or any other query mechanism?
- Are there any other types of software entity (apart from Annotation, Api, Class, Constructor, Enumeration, Exception, Interface, Library, Method, Package, Project) that you would like to search?
- Are there any other attributes of software entities (apart from type, programming language, license, keywords) that you would like to use in queries or see in the search results.
- Are there any other relationships between software entities that you would like to use in queries or see in the search results? (apart from inheritance, implementation, required interfaces, part-of/nesting, function/method invocation, import, revision history, static linking of library, dynamic linking of library, used/using projects, other(s): what types?

Questions related to browsing capabilities

- Do you have any suggestion on how to improve the way that MARKOS shows and/or allows to browse the software relationships (e.g. inheritance, implementation, required interfaces, part-of/nesting, function/method invocation, import, revision history, static linking of library, dynamic linking of library, used/using projects). How?
- When you have found a useful class (or method) would you just integrate in your software the single class, or rather the entire package/library? In other words, what is the level of granularity at which you will likely perform reuse? (*Entire library, Package contained in a library, Single class, or class cluster, Method contained in a class (assuming it can be isolated), Source code fragment*)
- When you have found a useful source code element, would you be interested to see if the same (or similar) source code element is also contained in other projects? Why?
- Suppose you are developing a Java software, and you search for a Java class/method that fulfills a given piece of functionality. Unfortunately, you did not find any solution implemented in Java. However, there are implementations in other languages (e.g., C++). Are they useful for you? Why?
- What are the premier sources of information to determine whether a method is suitable to fulfill a given piece of functionality?

Questions related to license analysis capabilities

- Do you have a basic understanding of copyright law, which is in your view sufficient for selecting open source licenses without professional legal advice?
- How important is it for you to be able check whether the Open Source license you have chosen for some project is compatible with the licenses of the components used by the project?
- Would a fully automatic check of licenses compatibility issues, based on a simplified, global model of copyright law, be useful to you, even though the check cannot take into consideration the specific copyright laws of your country?
- Do you typically consult a lawyer before selecting an Open Source license to use with your software? Does your company have its own legal staff to assist you with licensing issues?
- How much time do you typically need to analyse licensing issues and choose an open source license for your projects?
- Would you be interested in having the option to publish an analysis of the licensing issues of your software? Would you use this option to publish the analysis it shows that your project might have licensing issues?
- How much extra time training a would you (as a developer) be able to take to get more in-depth knowledge on license analysis compatibility argumentation?

Questions related to user comments capabilities

- Would you, as a software developer, be willing to provide feedback (reviews) about the quality of the information offered by MARKOS?
- Would you, as a software developer, be willing to provide feedback (reviews) about your experience with the projects/libraries you have retrieved from MARKOS?
- Would you, as a project manager, be willing to check user feedback and change the information about your project if necessary?

- Would you, as a project manager, be willing to give feedback to the reviews of MARKOS users concerning your project?
- Would you, as a project manager, be willing to improve the functionality of your project based on feedback from MARKOS users?
- Do you think that the exchange of information with other MARKOS users will improve the usability and benefits of MARKOS for your work?

Questions related to exploitation

- How does the prototype compare to the current site experience?
- To what extent can MARKOS impact your working routine?
- To what extent can MARKOS impact the quality of your work?
- Please estimate how many times a year you would use a system like MARKOS:
.....times a year
- Does MARKOS allow reducing the development cost/effort? If yes, please provide an estimate (in percentage)