

SEACW DELIVERABLE D.I.3

“Contribution to Standardization Report”

Project Acronym	SEACW
Grant Agreement No.	325146
Project Title	Social Ecosystem for Antiaging, Capacitation and Well-Being
Deliverable Reference Number	SEACW_D.1.3
Deliverable Title	Contribution to Standardization Report
Revision Number	1.0
Deliverable Editors <i>(main writers)</i>	UC3M ÁLIAD

Project co-funded by the European Commission within the ICT Policy Support Programme	
Dissemination Level	
RE	RESTRICTED

Revision	Date	Description
0	09/09/2013	Definition of objectives and expected results
0.1	18/11/2013	First document revision
0.2	11/12/2013	Second document revision
1.0	03/02/2014	Deliverable Version 1

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and work of others has been made through appropriate citation, quotation or both.

Abstract:

This document introduces the contributions proposed by the consortium who is taking part of the project for the improvement and standardization of European projects.

This document is the first of two deliveries. The current delivery takes place in the 12th month of the project, while the second will take place in the 24th month.

TABLE OF CONTENTS

1	Introduction.....	4
2	Standardization Report	5
2.1	Why a style guide is needed?.....	5
2.2	Style guide samples.....	6
2.2.1.	Web style guide index	6
2.2.2.	Apps style guide index.....	7
2.2.3.	Accesibility section	7
2.2.4.	Architecture of information – General Organization	8
2.2.5.	Architecture of information – Main Page	8
2.2.6.	Architecture of information – Videos Page.....	9
2.2.7.	Style – Learn Menu.....	9
2.2.8.	Style – Course Page	10
2.2.9.	Style – News Page	10
2.2.10.	Architecture of information – Apps Main Menu.....	11
2.2.11.	Style – Apps Main Menu	11
3	Social Developer.....	12
3.1	Context	12
3.2	Needs.....	13
4	Software Development Communities.....	14
4.1	Why a community?	14
4.2	Actions.....	16

1 Introduction

This document introduces the contributions for the standardization of European projects.

The consortium proposes the next contributions for standardization:

- ➔ A style guide document. This document explains why such a style guide should be a standard and displays some examples of the guide made for this project. The complete style guide is available in document D3.6 “The human interface requirements document” (appendix I).
- ➔ The creation of the concept of “social development”.
- ➔ The creation of an opensource community which is aimed at developing software for the scope of Active and Healthy Ageing.

With these proposals, the consortium intends to improve the execution of European projects and to carry out activities to promote the concept of Active and Healthy Ageing.

Since this is a first version of the Deliverable 1.3 which includes the preliminary ideas on the standardization processes to be carried out in this project. The second version that will be submitted on month 24th, will include the final guidelines that SEACW will share with the European Commission and with the community of technological volunteers that are willing to use them.

2 Standardization Report

For contributing to the standardization, SEACW consortium proposes to standardize a set of documents serving as guide for the future projects that will be developed.

The proposed document is the **style guide**. This document is especially important in a project aimed at developing software for social inclusion, as the outcomes themselves have very specific needs and thus, special care should be taken when studying the style and developing in the development phase of the project.

2.1 Why a style guide is needed?

We consider that, before development, it is very important to carefully analyse the scope and possible difficulties that final users could find.

For that reason, the style guide should be designed and written following the next steps:

- ➔ First, the concept of what is going to be developed must be clearly detailed, taking into consideration the target users and the objectives to be accomplished after the development.
- ➔ Design the structure for the development, what outcomes will be shown to users and what elements are common to all these outcomes.
- ➔ Study the accessibility guidelines which should be followed given the users and norms established by the W3C in the “Web Accessibility Initiative”.
- ➔ Design the architecture of information, a schema which details page by page where and when will be shown the information during the development, making mockups for the design and, finally, understanding what the final product will be.
- ➔ Define the style, where all the mockups made in the previous phase are analysed, and all the technical specifications which should be followed during the development are observed, such as:
 - Metrics and grids
 - Colour
 - Typography
 - States and feedback.
 - Iconography
 - Pictures and media

This document aims at facilitating the work of developers, such as the specification of the design and functionality of software products. Moreover, the communication between the different departments or partners taking part in the project gets simpler and efficient, as all of them already have a copy of the specifications of the final product before the development phases start.

Because of all the advantages described so far, the style guide provides many benefits during the development of a project, regardless its size, as it increases the efficiency of the different teams or partners, both in terms of communication and of technical work.

It is very important that this document is elaborated for all different devices and/or environments from which the development will be accessible.

2.2 Style guide samples

Next, a set of screenshots for the style guide developed for the SEACW Project are displayed.

2.2.1. Web style guide index

Index

Style guide

Web style guide

- 1. Introduction
 - 1.1. Concept
 - 1.2. Basic rules
- 2. Accessibility
- 3. Architecture of Information
 - 3.1. Permanent base layout
 - 3.2. General organization
 - 3.3. Home page
 - 3.4. Learn page
 - 3.4.1. Courses pages
 - 3.5. See page
 - 3.5.1. Videos pages
 - 3.5.2. News pages
 - 3.6. Interact page
 - 3.6.1. Forum pages
 - 3.6.2. Chat page
 - 3.7. Login / Join us page
 - 3.7.1. Sign-in process pages
 - 3.7.2. Error notification page
 - 3.7.3. User profile page
 - 3.7.4. Edit profile process pages
 - 3.8. Search pages
- 4. Style
 - 4.1. Metrics and Grids
 - 4.1.1. Permanent base layout
 - 4.1.2. Home pages
 - 4.1.3. Learn pages
 - 4.1.4. See pages
 - 4.1.5. Interact pages
 - 4.1.6. Login / Join us pages
 - 4.1.7. Search pages
 - 4.2. Color
 - 4.3. Typography
 - 4.4. States and feedback
 - 4.5. Iconography
 - 4.6. Pictures & Media
 - 4.7. Footer

2.2.2. Apps style guide index

Index

Style guide

Apps style guide

1. Introduction

- 1.1. Concept
- 1.2. Basic rules

2. Accessibility

3. Architecture of Information

- 3.1. Permanent base layout
- 3.2. App cover screen
- 3.3. Main menu
- 3.4. Secondary menu
- 3.5. Sections list
- 3.6. Features list
- 3.7. Contents window
- 3.8. Calendar
- 3.9. Form

4. Style

- 4.1. Metrics and Grids
 - 4.1.1. Permanent base layout
 - 4.1.2. App cover screen
 - 4.1.3. Main menu
 - 4.1.4. Secondary menu
 - 4.1.5. Sections list
 - 4.1.6. Features list
 - 4.1.7. Contents window
 - 4.1.8. Calendar
 - 4.1.9. Form

- 4.2. Color
- 4.3. Typography
- 4.4. States and feedback
- 4.5. Iconography



3

2.2.3. Accessibility section

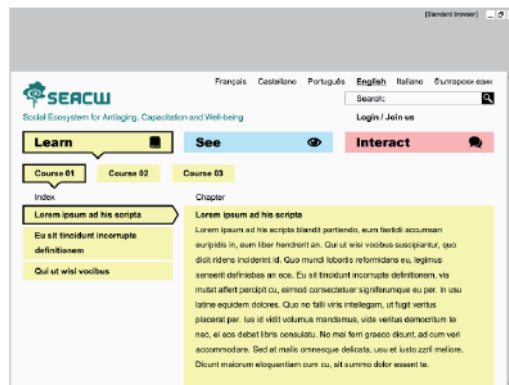
2. Accessibility

Web style guide

Colour and background

Colour is understood in the design design as a location reinforcement when navigating the ecosystem. Three main colors in light tones are used to reinforce the three different fields to be explored. Once the user enters one of the main three sections, its related colour will define the windows background of the secondary menu, the lists of contents and the content windows.

This three colour selection defines through its extensive use and pregnant light tones the visual identity of the project. The project needs not only achieving accessibility but expressing it as its main concern. The intense sharp contrast between the light colour backgrounds and the black arial type automatically delivers the message.



12

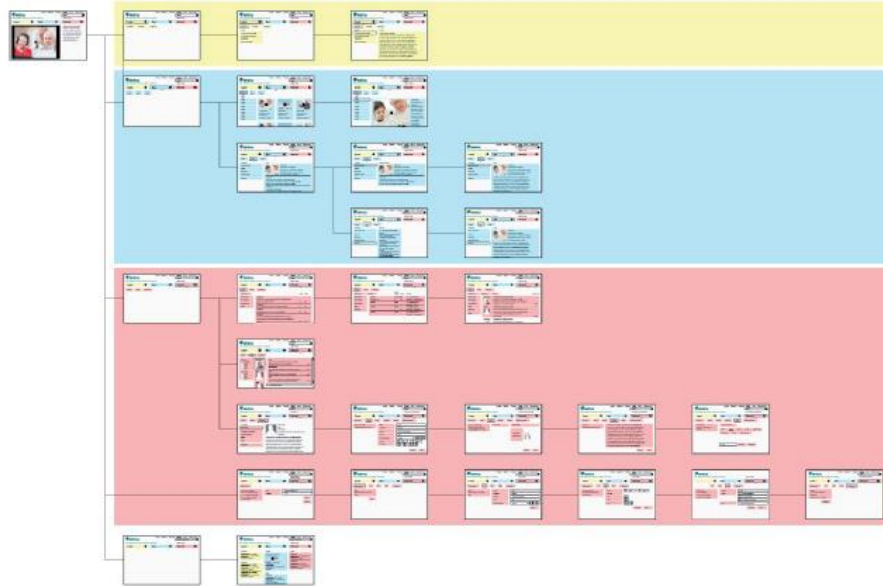
7

2.2.4. Architecture of information - General Organization

3. Architecture of information

Web style guide

3.2. General organization



2.2.5. Architecture of information - Main Page

3. Architecture of information

Web style guide

3.3. Home page

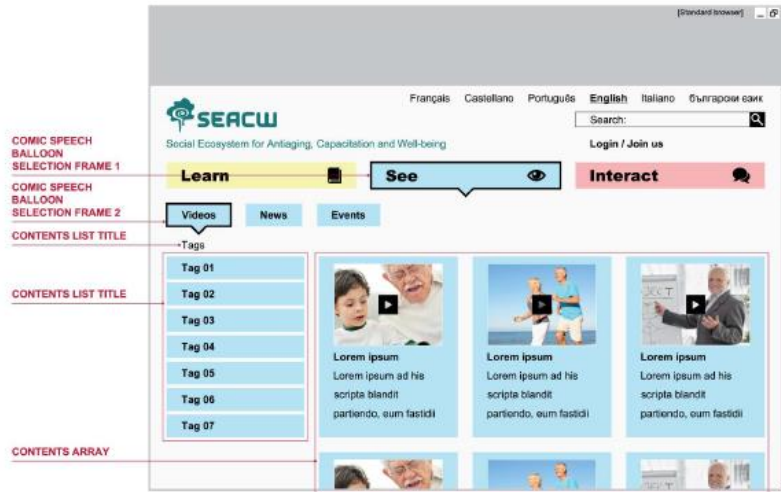


2.2.6. Architecture of information - Videos Page

3. Architecture of information

Web style guide

3.5.1. Videos pages



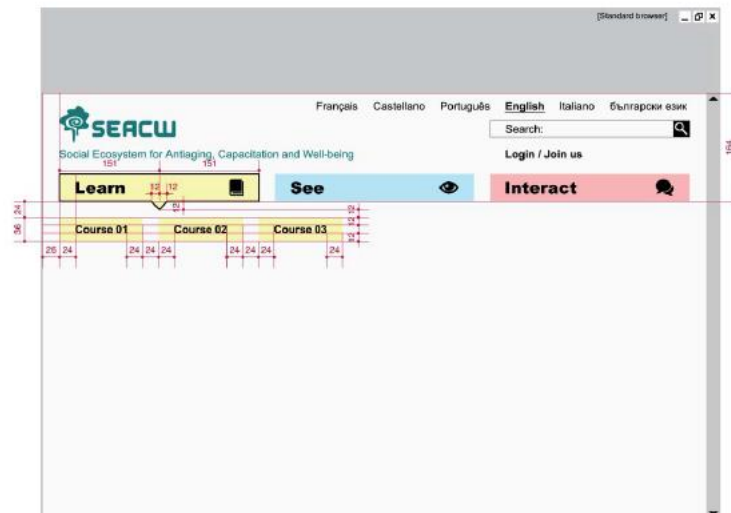
29

2.2.7. Style - Learn Menu

4. Style

Web style guide

4.1.3. Learn pages



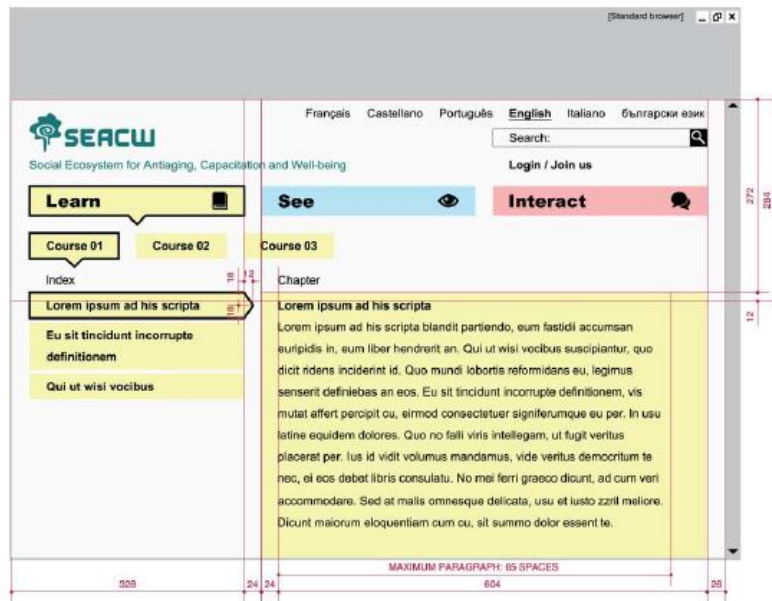
85

9

2.2.8. Style - Course Page

4. Style

Web style guide



87

2.2.9. Style - News Page

4. Style

Web style guide



94

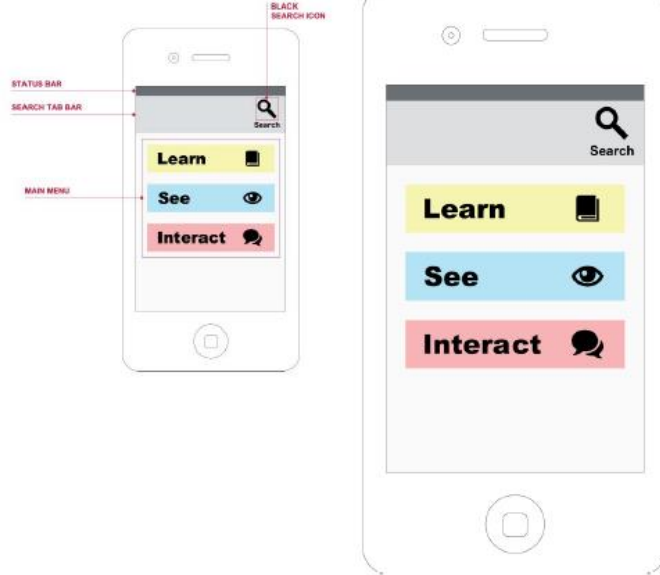
10

2.2.10. Architecture of information - Apps Main Menu

3. Architecture of information

Apps style guide

3.3. Main menu



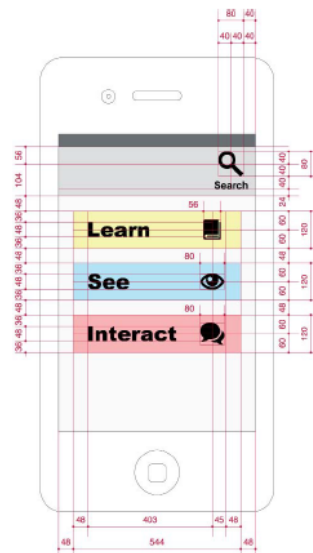
130

2.2.11. Style - Apps Main Menu

4. Style

Apps style guide

4.1.3. Main menu



139

3 Social Developer

SEACW Project is framed within a subset of European projects with a strong social nature, not only because of its topic, but also for its objectives, which are aimed at changing society, improving it and making it more human, relying on ICTs to achieve these goals.

This kind of initiatives could be framed within a volunteering movement which is appearing in different cities all over the world, also referred to as “technological volunteering”.

The objective being technological volunteering is to bring the New Information and Communication Technologies closer to everybody, and especially to those people who have higher handicaps to get access to those, or who are in risk of digital exclusion because of their lack of technological knowledge, thus reducing the inequality in the access to the Information Society and the Digital World.

Technological volunteering is a new, different way of becoming a volunteer. Any person with a certain minimum knowledge can turn into the perfect volunteer to help those who are taking the first steps to the use of new technologies. If you can find some spare time and you enjoy sharing your knowledge, you definitely should try this experience.

Once the concept of technological volunteering is established, we also want to go a step beyond and incorporate to this movement the concept of “social development”.



Figure 1. A volunteer with an elderly

3.1 Context

Within the technological world, during the last years different initiatives have arisen whose objective were to help those people who were in risk of being excluded of the digital society. A famous example would be the OLPC (One Laptop per Child) project, where technology is put to the service of society.

Additionally, other altruistic movements have also arisen, such as the Open Source movement where, with completely different purposes (in this case, pursuing the freedom of information in software), many resources and people time is spent in an unselfish manner, achieving spectacular outcomes.

Following the trail of this movement, we definitely think that a new role and a new term should be coined, which is built above the current concept of open source software, and should be called “social developer”.

The concept of open source was first used by the communities in 1998, trying to replace the ambiguous existing term of free software, where “free” could be understood either by the perspective of freedom or from the perspective of cost. For our project, it is intended to understand that the software can be used, read, modified and redistributed with no cost, and users and developers can use it freely.

The term was not accepted by some, considering that it removed the idea of freedom which could be found being “free software”, and could lead to confusion as the software could be understood just as free of cost. However, the new term still can be used by people who, even if they do not distribute the software free of cost, they deliver the source code so that it can be reviewed or modified by customers.

From now on, we will refer with the term “free software” to those pieces of software which are published with total freedom of use, modification and redistribution under the only explicit constraint of not modifying or restricting that freedom in the future.

From a semantic point of view, the literal meaning of “open source” is that the source code from a computer software can be observed, so it is clearly more restrictive and weak than the concept of “free software”. However, in most cases both terms are used to refer to the same set of licenses, keeping the same philosophy.

However, an explicit difference should be remarked between that open source software that enable users to modify and improve the software, and that software which only allows to visualize the code, restricting its use or modification.

Nowadays, the open source concept is also used to refer to a new software movement (the Open Source Initiative), which is different to the free software movement, as they both have different philosophical principles but with the same freedoms in practice.

The idea behind the concept is simple: when a community of developers can read, modify and redistribute the source code of a software application, then it evolves and improves. Users can adapt the software to their needs, can fix its bugs in very short periods of time and, eventually, produce better software.

3.2 Needs

There are many different social challenges which we confront nowadays. Most of us agree to a certain extent that Internet and the ICTs are a key element in the creation of resources for people. There are many different volunteering movements where people help learning the use of new technologies to people who don't have any technological background. Our proposal goes a step beyond and aims at providing people tools aimed at improving people wealth, and solving some of their daily problems.

There are different environments where a social developer could be useful:

- Helping disadvantaged communities to digitalize their cultural background
- Developing automatic translation tools for languages in minor use
- Adapting daily-use applications for handicapped people.

The list of possible environments and tasks could be unlimited.

4 Software Development Communities

4.1 Why a community?

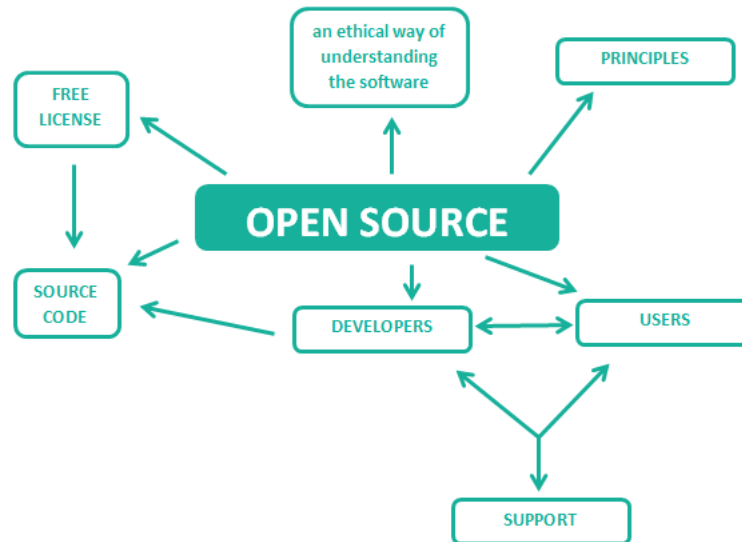


Figure 2. Open source structure

To a management extent, opensource communities are the best example. The functioning of these communities is based on the next features:

- ➔ Opensource software is an ethical way to understand software from the development, and also in the use and distribution.
- ➔ Opensource refers also to a significant set of values: ethics, creativity, efficiency, collaboration, equality, science, transparency, security, competitiveness, privacy, solidarity and, overall, freedom.
- ➔ Opensource software requires a free license which guarantee:
 - The right to publish improvements from which the community can take benefit.
 - The right to redistribute copies for anybody.
 - The right to adapt the program for each one's needs.
 - The right to use it with any (legal) purpose.
- ➔ Opensource software is distributed with the source code and is compiled with applications which respect open standards and free contents, such as fonts, translations, templates, multimedia, manuals, etc.
- ➔ Opensource software is designed for users and developers to cooperate among them, giving and receiving support.



Figure 3. Open source

While it is clear that a single developer can contribute to some extent to improve social inclusion and projects aimed at achieving it, it is really a team of developers who are aching to contribute who can produce significant benefit to achieve the goals for this kind of projects.



Figure 4. A social developer

For that to happen, it is required that a meeting point for all volunteers is designed so that the Open Source community of social developers can meet and share ideas with the ultimate purpose of achieving digital inclusion, and where they can communicate with each other and coordinate their efforts.

Of course, this community should establish some working and behaviour rules in order to favour the pace of the community and to encourage the cooperation between all the members in the community.

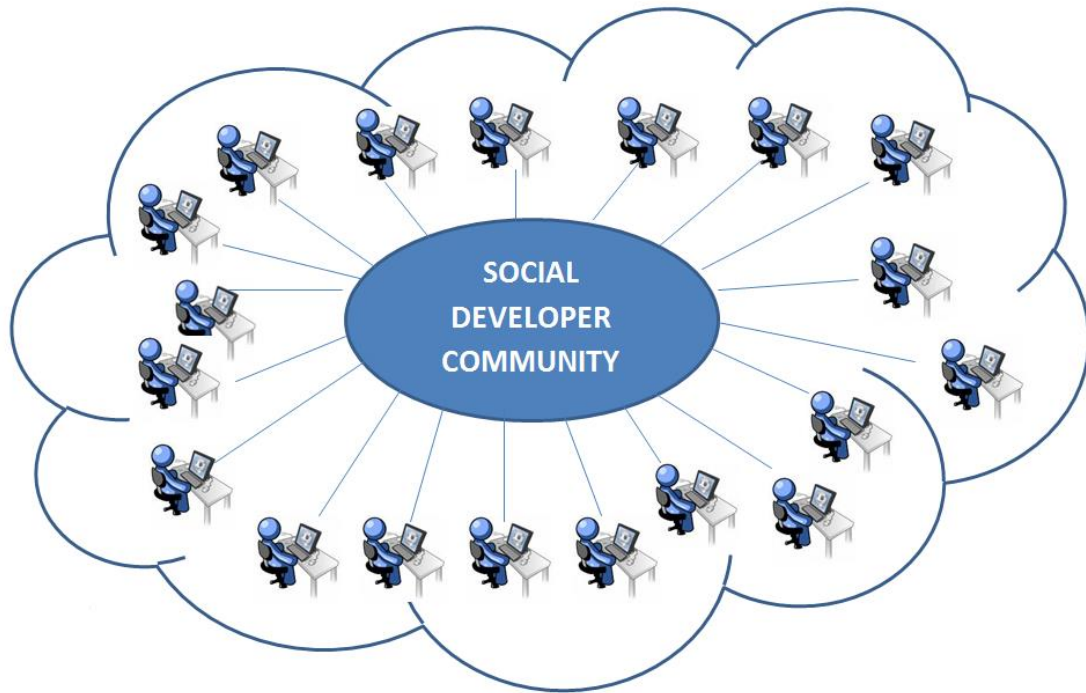


Figure 5. Social developer community

4.2 Actions

To create this community of social developers, a sequence of complex steps should be taken, which require technical, legal and promotional efforts. In a first phase, the next step could be enumerated:

1. Launching a website illustrating the concept of a social developer and his objectives.
2. Publishing software tools under opensource licence and a “social project” or “social developer” seal. The first projects could involve some of the tools of the ecosystem, and already existing opensource projects could be invited to take part.
3. Promote the concept and projects in social networks.
4. Define a procedure to evaluate projects in order to obtain the “social project” or “social developer” seal.