



| | |
|---|---|
| Project Acronym: | **ALERT** |
| Project Title: | **Active support and reaL-time coordination based on Event pRocessing in open source software developmenT** |
| Project Number: | **258098** |
| Instrument: | **Collaborative Project** |
| Thematic Priority: | **ICT-2009.1.2 – Internet of Services, Software and Virtualisation** |

# D9.9.1 – Exploitation Plan

*WP 9:* *Dissemination & Exploitation*

| | | |
|---|---|---|
| **Due Date:** | | M12 |
| **Submission Date:** | | 30/09/2011 |
| **Start Date of Project:** | | 01/10/2010 |
| **Duration of Project:** | | 30 months |
| **Organisation Responsible for the Deliverable:** | | Atos Origin |
| **Version:** | | 1.0 |
| **Status** | | Final |
| **Author(s):** | Belén Serrabou<br>Clara Pezuela Robles | Atos<br>Atos |
| **Reviewer(s)** | Dusan Zirojevic | CIM |

This work is partially funded by EU under the grant of IST-FP7-258098

| Project co-funded by the European Commission within the Seventh Framework Programme | | |
|---|---|---|
| **Dissemination Level** | | |
| **PU** | Public | **X** |
| **CO** | Confidential, only for members of the consortium (including the Commission) | |

# Version History

| Version | Date | Comments, Changes, Status | Authors, contributors, reviewers |
|---|---|---|---|
| 0.1 | 02/06/2011 | Initial ToC | Belén Serrabou (Atos), Clara Pezuela Robles (Atos) |
| 0.2 | 17/07/2011 | Added section 2.2, 2.3, 3, 4, 5, 6, | Belén Serrabou (Atos), Clara Pezuela Robles (Atos) |
| 0.3 | 22/07/2011 | Draft version sent to partners for contribution | Belén Serrabou (Atos), Clara Pezuela Robles (Atos) |
| 0.4 | 29/08/2011 | Contributions to section 3 | Clara Pezuela Robles (Atos), Felipe Ortega (URJC), Ljiljana Stojanovic (FZI), Predrag Stoiljkovic (CIM), Bratislav Stoiljkovic (CIM), Dusan Zirojevic (CIM), Balázs (Corvino), |
| 0.5 | 31/08/2011 | Sections 4 and 6 | Clara Pezuela Robles (Atos) |
| 0.6 | 02/09/2011 | Integration of first round of contributions | Clara Pezuela Robles (Atos), Belén Serrabou (Atos), Felipe Ortega (URJC), Ljiljana Stojanovic (FZI), Predrag Stoiljkovic (CIM), Bratislav Stoiljkovic (CIM), Dusan Zirojevic (CIM), Balázs (Corvino), |
| 0.7 | 08/09/2011 | New integration of contributions | Dimitris (ICCS), Stuart Javis (KDE), Felipe Ortega (URJC), Belén Serrabou (ATOS), Clara Pezuela (ATOS) |
| 0.8 | 15/09/2011 | Open Source Communities market analysis. Revision and improvement of section 4. | Mathieu Le Breton (EBM), Stuart Jarvis (KDE), Belén Serrabou (ATOS), Clara Pezuela (ATOS) |
| 0.9 | 21/09/2011 | Internal review | Dusan Zirojevic (CIM) |
| 1.0 | 30/09/2011 | Modifications accordingly and Ready for submission | Belén Serrabou (ATOS), Clara Pezuela (ATOS) |

## Table of Contents

## Index of Figures



## Index of Tables

## Executive Summary

The objective of this deliverable is to provide an initial version of the ALERT exploitation plan. First of all we analysed the market. Although the conceptual idea of the project was thought focused on Open Source Communities, the further analysis of potential users has concluded in considering also the software industry as a relevant market for ALERT as well. Regarding Open Source Communities we have analysed the most famous platforms to host open source projects such as SourceForge, Google code project hosting, GitHub...etc. The total number of active open source project was estimated in 18 000 active projects in 2007 by Roberto Galoppini, in FLOSSmetrics project [14]. Since this was in 2007, today we could think about 20,000 to 50,000 active projects. There is a clear need for the ALERT software in open-source communities, however, that does not mean that there will be no resistance to its integration. ALERT has some highly desirable attributes built in from the start. It is developed as open-source software, and it is being developed with input and cooperation from well-known open-source communities.

Regarding the software industry, OSS has been a disruptive force in the IT industry for well over a decade. Its growth has been tremendous and seems not being affected by the economic crisis. OSS is perceived by the market and the end-users as a very cost efficient software delivery mode, which is a real advantage in these troubled times. PAC Consultants had estimated the potential growth for the OSS Market in Europe (EU27) to reach more than 350.000 million of Euros by 2020. Recently, we have also seen OSS solutions take a more direct role in new, emerging and innovative market opportunities as well ( e.g. cloud computing, web mobile, extreme transaction processing, etc.). Main limitations when it comes to resolving bugs in the main Open Source Communities are lack of user-visibility over the code structure, complex interactions between teams, duplication of bug reports... these are the problems ALERT is trying to solve.

ALERT, by producing a solution for the better open source development process, will have a strong impact on the increasing use of OSS. The ALERT target users are: Open Source communities, large companies and SMEs developing OSS and the Research Community.

ALERT goal is to create an active collaboration platform (i.e., a virtual actor that interacts with the developers) that aims at identifying different interactions, suggesting actions and bringing past interactions into the developers' attention enabling developers to work better together. For that, ALERT extracts information from different sources from developers' environments (mail, forums, chat, bug reports, code repository, etc.) using a set of sensors installed in these community tools. Then, it analyses the collected information extracting metadata, annotating the data and storing it by using an ontology. The implementation of this project idea will provide the following type of results:

- Conceptual approach
- Use cases
- ALERT ontology
- Software components
- Pilots
- Scientist publications

All results have been identified and analysed in this document.

The consortium has been doing a brainstorming in order to ensure the ALERT sustainability after the end of the project. As our key target is used to work in

communities, the ALERT consortium has decided to **create a community of interest around ALERT**. It will be a toolset repository, where all components developed in the project will be available to download. In addition, the web site will be improved and it will include forums with related hot topics to discuss, articles, documentation, publications, blog, news, etc.

The first decision to develop the sustainability plan of ALERT is to decide how ALERT will be delivered. After analysing different options the consortium decided that the most appropriate option for ALERT results is to release the components individually, although other choices could be considered as well.

The second phase is to define the business model. We have identified two different business models, on one side there are the project partners or industrial companies advertised in the community, and on the other side, there are the business models of the community itself in order to guarantee the sustainability of the community.

The third phase is to define the community strategy. The **main goal** of the community building is to reach the sustainability of ALERT results and their evolution beyond the project life. There are several tasks to be undertaken to create a community, for the moment we identified the following:

1. Market research
2. Content provider
3. Paid advertising
4. Link building
5. Branding
6. Social Networking
7. Monitoring

The tasks will be distributed among the partners taking into account the effort available, the core competencies of the partners and their role within the project. The tasks will be analysed during the next period of the project. We have identified the next steps to build the ALERT community. Creating a community entails higher risk, but also higher impact when succeeds.

The consortium has analysed the IPR and licensing as an essential task in the exploitation work package. All the project documentation has been delivered under Creative Commons licenses:

- Attribution-no derivatives (BY-ND) for dissemination material

- Attribution-Share alike (BY-SA) for the public deliverables (all except the management reports and exploitation plans).

In the case of software results, the licensing issue has required a deeper analysis and work explained further in this document. A work plan has been established to proceed with a licensing analysis. The final outcome can be found in the annex, as a summary of the adopted licenses:

- 6 components under GPL v2
- 4 components under LGPL v3
- 2 components under BSD

The last section in this document is addressed to the individual exploitation plans for the ALERT partners. They have different interests on what ALERT will produce and due to their respective nature (industry, academic, research or community) their business goal is completely different and the benefits they expect from the project as well. We have classified the exploitation plans according to this nature and the requested information is slightly different. These plans can evolve during the project and an updated version will be provided at the end of the project.

# 1 Introduction

The ALERT exploitation plan is included in the WP9 Dissemination and Exploitation. The exploitation part is divided in four different tasks, distributed in two deliverables. The first deliverable contains the identification of results and market analysis, and targeting and positioning tasks. The second deliverable will include the competitive analysis and business planning tasks.

We have defined the ALERT exploitation strategy through the duration of the project, and it has the following phases:


**Figure 1 Phases of ALERT exploitation strategy definition**

Following on the Executive Summary, the first section of the deliverable is the introduction, where the ALERT exploitation strategy is defined through the duration of the project. Section 2 describes a market analysis of target sectors, Open Source Communities and Software industry. First of all we analysed the market size to ensure we will produce a greater return. The OSS players are examined and the current trends and limitations of bug resolution in OSS are considered to envisage the current needs of the market. The target market is defined to better address the ALERT results and focus on the dissemination activities through these sectors.

Section 3 identifies and describes all the ALERT results. The ALERT software components include the problem solved, technical innovation and added value.

Section 4 tackles the sustainability plan. It describes the results of the brainstorming about sustainability and the plan decided by the consortium. It describes the different phases to develop the ALERT sustainability plan.

Section 5 describes an initial SWOT analysis.

Section 6 examines a preliminary IPR and licensing analysis of ALERT results.

The last section of the deliverable is devoted to the individual exploitation plans. Every ALERT partner has drafted their initial plans focusing on the identification of ALERT business opportunities, the potential addressable market and customers, customer needs and value proposition. In the next deliverable the project partners will present their final individual exploitation plans.

## 1.1 Purpose

The purposes of this deliverable are as follows:

- Develop a market research of the most relevant sectors
- Identify the ALERT target market
- Describe the ALERT results
- Outline the sustainability plan of the ALERT project
- Provide an initial individual exploitation plan of all the project partners
- Provide a preliminary IPR and licensing analysis

## 1.2 Glossary of Acronyms

| Acronym | Definition |
|---|---|
| BSD | Berkeley Software Distribution |
| D | Deliverable |
| DRS | Document Review Sheet |
| EC | European Commission |
| FLOSS | Free/libre/open-source software |
| GPL | General Public License |
| LGPL | Lesser General Public License |
| OSS | Open Source Software |

| | |
|---|---|
| **PM** | Project Manager |
| **PO** | Project Officer |
| **SaaS** | Software as a Service |
| **SSBS** | Software and Software-Based Services |
| **SWOT** | Strengths, Weaknesses, Opportunities, Threats |
| **WP** | Work Package |

# 2 Market Analysis

The ALERT results can fit into different types of markets (users). Although the conceptual idea of the project was thought to be focused on Open Source Communities, the further analysis of potential users has concluded in considering also the software industry as a relevant market for ALERT as well.

Thus, this section describes the two big markets we address: Open Source Communities and Software Industry, to introduce next the limitation of the bug resolution currently experienced by developers and finally to enumerate the concrete target markets for which we will define our strategy of exploitation.

## 2.1 Open Source Communities

Given that the Open Source Communities do not represent a traditional market as it is understood in business world, they are the main target audience of ALERT results. In fact, the project idea rose from the current needs of Open Source developers and how to satisfy them. The way to exploit ALERT in Open Source communities will be diverse and depending on partners' strategies, but it can be from the dissemination of the results for the adoption by as many communities as possible, to the commercial offering of advanced services by ALERT partners to some interested community.

Therefore this section describes the Open Source Communities market, providing figures about size, projects, etc. It points out which are the main Open Source players, the most important OSS communities: their characteristics, their current limitation of problems to identify their market needs.

### 2.1.1 Projects topology of famous projects

Several types of open-source projects exist, with example of famous projects below [14]. They might be business-oriented, supported mostly by companies like Petals ESB, or by a mixture of paid and volunteer individuals, like KDE. Large projects often include lots of independent sub-projects. KDE, for example, began as a highly focussed effort to build a new desktop for Linux, but has developed into a wide community encompassing desktop software, supporting libraries, web-only projects and low-level infrastructure projects.

| Size/Hybridity | Volunteer | Mixed | Company |
|---|---|---|---|
| **Small** | Wordpress | | MySQL |
| **Medium** | OpenBSD | Mozilla | OpenSolaris |
| **Large** | Debian | Linux (kernel) | Eclipse |

**Table 1 Topology of OS projects**

Total number of active open source projects was estimated in **18 000 active projects in 2007** by Roberto Galoppini, in FLOSSmetrics project [15]. Since this was in 2007, this number might have grown up even more, and today we could think about 20 000 to 50 000 active projects.

### 2.1.2 Major open source project communities

Below is an overview of most famous platforms to host open source projects. There are very different types of platforms. Some might be around providing one coherent package, like Eclipse or Mozilla. Some might be very general, like Sourceforge, Github or Codeplex. Others, such as KDE, fall somewhere in the middle, with applications that work well together, but which can be used – and are created - in a highly modular fashion They might be hosted by companies (Google, Microsoft, Geeknet...) or foundations (Apache, KDE...)

Please note that the number of projects is not relevant to compare community's size, because projects might have very different granularities, and because some older provider might have a lot of inactive projects. The goal of this number is to show these differences in topologies.

| Platform name | Total number of projects | Owner of the website | Thematic |
|---|---|---|---|
| **SourceForge** | 310 000 repositories | Geeknet | General |
| **Google code project hosting** | 5500 projects | Google | Web/Mobile |
| **GitHub** | ~108 000 forked repositories [1] | GitHub | General |
| **Codeplex** | 25 000 projects | Microsoft | General |
| **FSF directory** | 6 600 packages | Free Software Foundation | General |
| **Apache** | 158 projects | Apache Foundation | Middleware |
| **OW2** | 130 projects | OW2 (French organization) | Middleware |
| **Eclipse** | 260 projects | The Eclipse Foundation | Development tooling, around Eclipse platform |
| **KDE** | 47 projects | KDE e.V (German non-profit) | Mainly end-user software on Linux, but also support libraries |
| **Mozilla** | 19 Big projects | Mozilla Foundation/ Mozilla corporation | Web clients, around Firefox |

**Table 2 Main OS projects**

[1] Because Git structure is different and based on forks, we need to evaluate it differently. Over 2 000 000 repositories in April, 45% were empty : https://github.com/blog/841-those-are-some-big-numbers and over 135 000 repositories, 12 000 (9%) were forked at least once : http://wiki.dandascalescu.com/essays/pita-threshold. Today there is 2 700 000 projects. 2 700 000 * 45% * 9% = 108 000

- SourceForge

SourceForge is a web-based source code repository. It acts as a centralized location for software developers to control and manage open source software development. The website runs a version of SourceForge Enterprise Edition, fork ed from the last open-source version available. As of July 2011, the SourceForge repository hosts more than 300,000 projects and has more than 2 million registered users, although not all are active.

- Google code project hosting

Google Code runs a project hosting service that provides revision control offering Subversion, Mercurial and Git (transparently implemented using BigTable as storage), an issue tracker, a wiki for documentation, and a file download feature. The service is available and free for all OSI-approved Open Source projects (as of 2010, it is strongly recommended but no longer required to use one of the nine well-known open source licenses: Apache, Artistic, BSD, GPLv2, GPLv3, LGPL, MIT, MPL and EPL). The site limits the number of projects one person can have to 25. Additionally, there is a limit as to the number of projects that may be created in one day.

- GitHub

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both commercial plans and free accounts for open source projects. The site provides social networking functionality such as feeds, followers and the network graph to display how developers work on their versions of a repository. GitHub also operates a pastebin-style site called Gist, wikis for individual repositories, and web pages that can be edited through a git repository. GitHub is operated under the name GitHub, Inc.

- Codeplex

CodePlex is an open source project hosting website from Microsoft. It allows shared development of open source software. CodePlex is an open-source Website where engineers and computer scientists share projects and ideas. Its features include wiki pages, source control based on Mercurial, Team Foundation Server or Subversion, discussion forums, issue tracking, project tagging, RSS support, statistics, and releases. Since January 22, 2010, Mercurial distributed source control system is supported as well, and this support has been enhanced.

- Free software directory

The Free Software Directory is a project of the Free Software Foundation (FSF) and UNESCO. It catalogues free software that runs under free operating systems - particularly GNU and Linux. Unlike some other directories that focus on free software, Free Software Directory staff verifies the licenses of software listed in the directory. The directory lists 6,600 packages.

- Apache

The Apache Software Foundation (ASF) is a non-profit corporation (to support Apache software projects, including the Apache HTTP Server. The Foundation recently accepted stewardship of the well known OpenOffice.org codebase, previously led by Oracle. The Apache Software Foundation is a decentralized community of developers. The software they produce is distributed under the terms of the Apache License and is therefore free and open source software (FOSS). The Apache projects are characterized by a collaborative, consensus-based development process and an open and pragmatic

software license. Each project is managed by a self-selected team of technical experts who are active contributors to the project.

- OW2

The OW2 Consortium is a not-for-profit consortium mainly devoted to producing open source middleware. It was founded by INRIA, Groupe Bull, and France Télécom. OW2 aims at developing open-source components for distributed applications (Web applications, grid computing, clusters, business integration, nomadic systems, etc). At the end of 2009, OW2 merged with the Open Solutions Alliance (OSA), a primarily U.S.-based consortium of open source software companies with the mission of fostering the adoption of open source software in enterprise level companies.

- KDE

KDE is a loose affiliation of contributors, creating open-source software primarily for the Linux desktop (however, KDE applications also run on Windows, Mac OS X and other Unix platforms and KDE also develops low level software supporting the desktop applications. It is supported by the German non-profit KDE e.V. Contributors are either volunteers or are employed by companies that have businesses built on the use of KDE software or interests in KDE technologies. For example, KDAB provides commercial support and expertise in groupware systems that make use of KDE's personal information management software and employ people to work on this. There are hundreds of KDE applications, some worked on by a single contributor, others the work of teams of tens of people.

- Mozilla

The Mozilla Foundation is a non-profit organization that exists to support and provide leadership for the open source Mozilla project. The organization sets the policies that govern development, operates key infrastructure and controls trademarks and other intellectual property. It owns two taxable for-profit subsidiaries: the Mozilla Corporation, which employs several Mozilla developers and coordinates releases of the Mozilla Firefox web browser, and Mozilla Messaging, Inc., which primarily develops the Mozilla Thunderbird email client. The Mozilla Foundation was founded by the Netscape-affiliated Mozilla Organization, and was officially launched on July 15, 2003. The organization is currently based in the Silicon Valley city of Mountain View, California, USA.

Mozilla foundation is famous for developing Firefox web browser, but also hosts other application projects, 8 overall: Firefox for desktop, Firefox for Mobile, Thunderbird, SeaMonkey, Lightning, Camino, Bugzilla, and Firebug.

- Eclipse

The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the open-source Eclipse Projects and helps cultivate both an open source community and an ecosystem of complementary products and services. The Eclipse project develops development tooling around Eclipse platform.

### 2.1.3 Diffusion to communities

Project topology must be taken into account, to diffuse any new developing tool, like ALERT.

- Big community: They include lot of sub-projects. Each sub-project being quite independent, it is harder to diffuse a new tool: It has to be accepted by each sub-project one by one. This is the case of OW2 and KDE
- Small community: Any change will be easier in a small community, and adopting a new tool should be easier, and need to convince just a few people. This is the case of Morfeo Community. However, the problems ALERT seeks to solve may be less acute in small communities with fewer issues to track and contributors to coordinate
- Funded by companies: They have more money than volunteers. They are eager to spend money for a new tool if they see value in it. They tend to choose more mature solutions, and as their employee time is valuable, they can spend money for it. As companies are hierarchical, a few people need to be convinced in order to use the tool. The end decisions whether to use a new tool or not belongs to the hierarchy, and is then more or less imposed to everyone.
- Funded by volunteers: They have little money, and would probably not buy any developing tool. They will choose open source software and implement it themselves as much as possible, avoiding to buy service, even if it takes them a lot of time. This may be as much a decision of principle as one of financial considerations. The choice of any new tool might trigger public discussions before adoption (the bigger the community, the longer the discussions)

### 2.1.4 Limitation and problems

There is a clear need for the ALERT software in open-source communities, as discussed below. However, that does not mean that there will be no resistance to its integration.

ALERT has some highly desirable attributes built in from the start. It is developed as open-source software, so can be adapted to a particular community's needs. It is also being developed with input and cooperation from well-known open-source communities. This enables useful feedback to be gathered during the development process and means ALERT should be easy to integrate into those communities from the start.

Open-source communities that are based mainly on volunteer contributions often have little hierarchy. In KDE, for example, there is no one who can impose the use of the ALERT software. It can be set up on a KDE server with the approval of KDE e.V. and integrated in the KDE issue tracker with the agreement of the KDE system administrators. However, individual contributors cannot be compelled to make use of the information ALERT offers, nor to provide the information it needs.

Even in more hierarchical communities, in which a leader or manager may be able to mandate the use of ALERT, a volunteer can simply choose to leave (or fork) the project.

Therefore, ALERT must quickly show its usefulness to persuade contributors that they should make use of it. This requires the ability to populate the system with useful information in advance, so that it can provide clearly useful services such as identification of duplicate issue reports. It also requires a high degree of configurability so that individual contributors can make it fit into their existing work-flows. This could take the form, for example, of enabling users to fine-tune the notifications they receive and the manner in which they receive them.

Integration of ALERT into work-flows will take time and it may be resisted by some individuals and teams. It should therefore be usable as a useful supplement to existing systems during its introduction. Its initial usefulness should also not be dependent on widespread adoption. For example, the system should be able to identify duplicates in issues reported to two teams, even if only one team is actively using ALERT.

### 2.1.5 Trends and needs of open source communities

#### *2.1.5.1 Volunteer Driven Communities Have High Contributor Turnover*

Volunteer-driven open-source communities can grow rapidly. There is a low barrier to entry (often no formal acceptance procedure – volunteers can simply begin submitting contributions). However, there can also be little commitment if the contributors are not paid. If contributing to the project stops being fun or the contributors suddenly lack time, they can leave the project easily. This tends to lead to a high turnover of contributors in many volunteer-driven open source communities.

In many ways, this is one of the strengths of open-source software. No single contributor is vital and the project can survive losing contributors. New contributors bring new ideas and the balance of power in decision making tends to rest with the most active contributors, so stagnation is prevented.

However, high turnover of personnel can also make it difficult for community members to form close working relationships with, or even good awareness of, people outside of their immediate team. As a result, there is a high reliance of communication structures such as mailing lists and issue tracking systems. This means that limitations of such systems can harm productivity as a lack of communication between teams – or simply a lack of knowledge of who is the best person to contact – can prevent the identification of contributors best suited to a particular task and result in needless duplication in resolving issues.

Loss of contributors over time (as they become too busy to contribute or their interests change) also makes it essential to continuously attract new contributors. Mailing lists and issue trackers can be hard for new contributors to navigate and understand.

ALERT can help with some of these issues, by making the distribution of expertise and relationship between issues experienced by multiple teams more apparent and by providing pointers towards the distribution of expertise within a community.

#### *2.1.5.2 Commercial-led communities may have hierarchies that prevent inter-team relationships*

In commercial-led communities, where most contributors are employed, growth is likely to be lower as it depends on available budgets and demand for products. However, the payment of a salary makes loss of contributors due to competing demands on time less likely, so communities can be more stable. However, this does not necessarily help contributors' awareness of other community members as they are likely to be in well-defined teams and answer to a senior member of the community.

ALERT can help in this case as managers should be able to use the system to understand their communities and identify other managers and teams with which they should liaise on issues and features.

As the use of electronic communications and version control systems becomes ever more widespread, communities can become more distributed in geographical location

and time zone, also preventing the building of strong personal relationships. The assistance of technology in understanding communities is therefore increasingly vital.

#### *2.1.5.3 Increasing Acceptance of Open-Source Software Brings New Types of User and Contributor*

Open-source software is no longer found only in niche applications. The popularity of the Mozilla Firefox web browser, Android mobile software and media applications such as VLC has brought many new users to open-source software who does not understand its development process. New users are welcome – they help to create markets for commercial exploitation and provide wider testing and a higher profile to open-s software. However, they have little experience of creating useful reports of software issues or even understanding to who an issue should be reported. This can lead to an unmanageable number of badly assigned, duplicate or poorly described issue reports in projects' bug trackers.

The ALERT software should be able to guide users in finding the relevant place to report their issue, automatically identify duplicates and add useful additional information by linking similar reports. This will enable open-source projects to continue to gain more users from the software but not be overwhelmed by new issue reports.

## 2.2 Software Industry

The software industry has evolved from their traditional segmentation. Today, software is increasingly the part of modern life. Many definitions exist when it comes to defining what software is. A recent study on “Economic and Social Impact of Software & Software-Based Services”, by Pierre Audoin Consultants [2] defines the Software and the Software-Based Services (SSBS) in four types of software products: applications, tools, system infrastructure software and games. The overall SSBS market – 228.6 billion Euros in 2008 – is expected to grow at a 3.7% CAGR by 2012 to reach 264.8 billion Euros [2]. The figure below shows the forecast development of the SSBS market in Europe (EU27) for the 2007-2012 period.

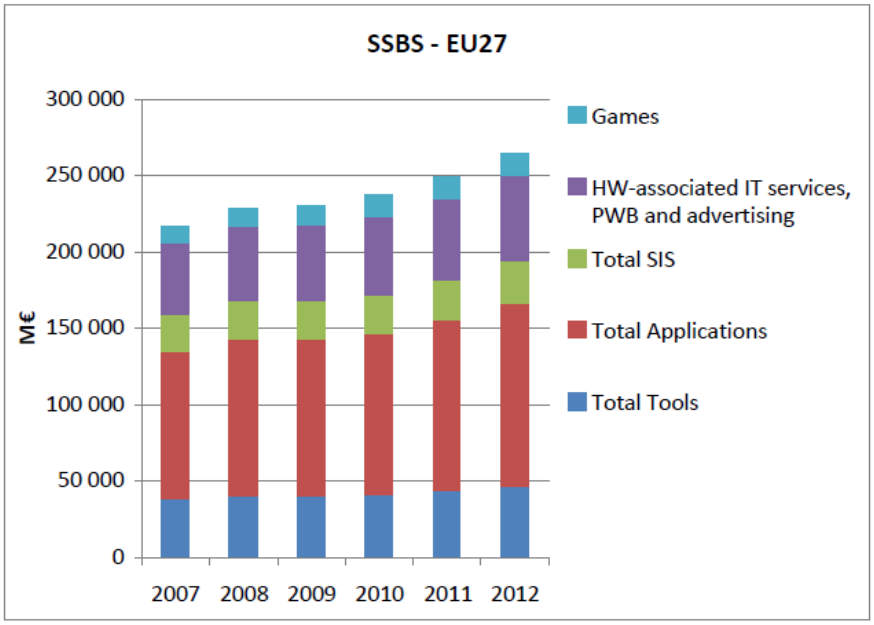


**Figure 2 SSBS market – EU27- by software types [2]**

The following table summarizes the main development characteristics of the SSBS market by revenue models for the 2007-2012 period.

| | Market Structure 2008 | Market Structure 2012 | CAGR 08/12 |
|---|---|---|---|
| Licenses + standard Maintenance | 29.0% | 26.7% | 1.9% |
| Associated IT services | 66.9% | 63.9% | 2.8% |
| Paid web-based | 2.1% | 4.2% | 18.7% |
| Advertising | 2.0% | 5.3% | 23.6% |
| Total SSBS | 100% | 100% | 3.7% |

**Table 3 SSBS market by revenues models 2007-2012 [2]**

The major rise will be on the advertising model, increasing at CAGR of 23,6% in only four years. The development of the paid web-based models will have also a major impact on the structure of the SSBS market. The Associated IT services will increase notably with respect to licenses + maintenance. Stronger development of OSS technologies would mean additional growth in the Associated IT services, as enterprises and administration would require supplementary integration services.

Open-source software has been a disruptive force in the IT industry for well over a decade. The disruptive potential of open-source software is far from exhausted, as new providers leverage it to expand the availability of solutions and delivery models. Its growth has been tremendous and seems not being affected by the economic crisis. OSS

is perceived by the market and the end-users as a very cost efficient software delivery mode, which is a real advantage in these troubled times.

PAC Consultants had estimated the potential growth for the OSS Market in Europe (EU27) according to their revenue models, 2008-2020 (millions Euros) as you can observe in the figure below:

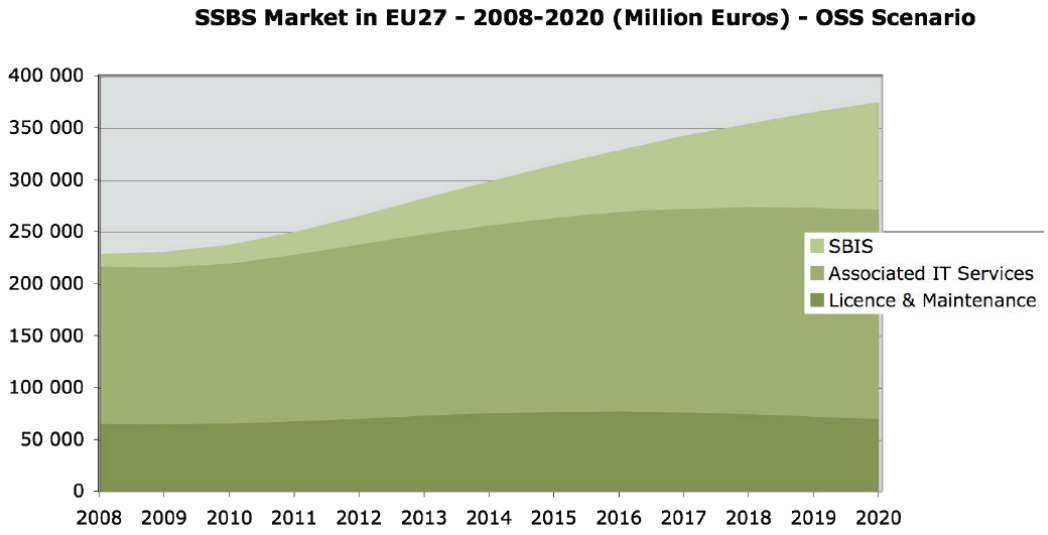


**Figure 3 OSS Market in EU27 – 2008-2020 [1]**

The market size of the OSS market is expected to grow increasingly. Due to the economic crisis, the OSS is already benefiting for their lower cost of ownership. It is predicted a slow growth of Licenses and Maintenance model, due to the traditional software products replaced by OSS, and indirectly, the higher diffusion of OSS technologies will mean stronger price pressure on the traditional software markets. The associated IT services will increase as the integration services will expand.

### 2.2.1 OSS Industrial Players

As the global economy recovered, worldwide operating system (OS) revenue totalled $30.4 billion in 2010, a 7.8 percent increase from 2009, according to Gartner, Inc.

Microsoft held the largest market share of the worldwide OS software segment, with 78.6 percent share in 2010 (see Table 2). The distant second- and third-place vendors were IBM and HP, with 7.5 percent and 3.7 percent share, respectively. Oracle climbed up the ranking from No. 8 in 2009 to No. 4 in 2010 by acquiring Sun Microsystems' Solaris business in April 2009.

| Vendor | 2010 Revenue | 2010 Market Share (%) | 2009 Revenue | 2009 Market Share (%) | 2009-2010 Growth (%) |
|---|---|---|---|---|---|
| Microsoft | 23,848 | 78.6 | 21,926 | 77.9 | 8.8 |
| IBM | 2,284 | 7.5 | 2,163 | 7.7 | 5.6 |
| HP | 1,125 | 3.7 | 1,109 | 3.9 | 1.4 |
| Oracle | 780 | 2.6 | 10 | 0.0 | 7,682.8 |
| Red Hat | 610 | 2.0 | 517 | 1.8 | 18.0 |
| Apple | 520 | 1.7 | 449 | 1.6 | 15.8 |
| Others | 1,183 | 3.9 | 1,968 | 7.0 | -39.9 |
| Total | 30,350 | 100.0 | 28,142 | 100.0 | 7.8 |

Source: Gartner (April 2011)

**Table 4 Worldwide OS Revenue and Market Share by Vendor (Millions of Dollars)[3]**

Red Hat has achieved clear leadership in the open-source enterprise application server market, with the largest installed base and partner following. Cloud and virtualization are its opportunities, while moving unpaid users to commercial support remains a consistent challenge.

### 2.2.2 OSS Trends

As The 451 Group stated [7] “Established vendors have adopted open source and cloud computing has changed the rules about how businesses adopt technology. The long-term opportunities are not in open source products and services, but products and services that are built on open source.”

Recently, we have also seen OSS solutions take a more direct role in new, emerging and innovative market opportunities as well ( e.g. cloud computing, web mobile, extreme transaction processing, etc.). In 2010, the Hype Cycle produced by Gartner [9] shows open-source technologies continuing to play a crucial role in multiple IT initiatives, as presented in the figure below:

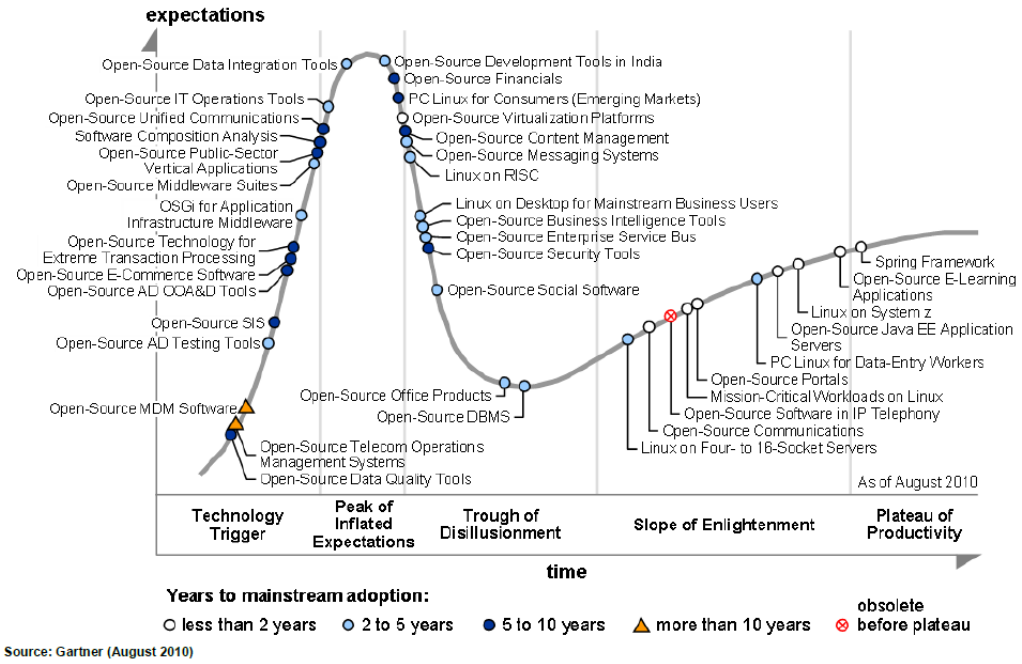


**Figure 4 Hype Cycle for Open-Source Software, 2010**

The most consistent distribution is further evidence of the growing maturity of open-source solutions and broad industry penetration. In this year's Hype Cycle, 77% of the technologies are rated at moderate or high impact. This is evidence of open source's impact on software markets at a relatively high business value, rather than a niche phenomenon.

One of the Gartner predictions 2011 regarding OSS is "Strategic Planning Assumption: By 2014, those organizations with effective, open-source community participation will consistently deliver high returns from their open-source investments".

This assessment reinforces the ALERT objectives and recognizes the real need for a solution as ALERT results.

There are several ways how industry uses OSS. A recent study provides a state of the art concerning integration of Open Source into industrial systems (a European research project called Qualipso [8]). These ways are:

- Internally: for infrastructural purposes, e.g. Linux, Apache, MySQL
- Internally: standalone-applications and tools, e.g. CVS, BugZilla
- In products: small components, e.g. libraries
- In products: large components, e.g. embedded Linux, MySQL
- For customers: in consulting services.

## 2.3 Limitations of bug resolution in OSS

One of the recommendations for 2011 from Gartner regarding OSS:

"Open-source software must move beyond a tactical sourcing option and become a strategically applied, in-house development methodology supported by software configuration and application life cycle management toolsets." [6]

It is clear that there is a room for improving the development of OSS software within communities or organizations. This will lead to a growth in products that support software development.

Main limitations when it comes to resolving bugs in the main Open Source Communities are [4]:

- Lack of user-visibility over the code structure causes that sometimes the users report a bug in respect to the wrong component or library because the bug effect appears to affect this component instead of the real one which has the bug.
- Complex interactions between teams. Distributed, multilingual and multidisciplinary developers spread all over the world face a complex scenario of interactions.
- Duplication of bug reports. Different users of the code detect the same bug and report it several times.
- Duplication of effort in downstream bug trackers. Many developers can be solving the same bug at the same time without interaction among them, duplicating the effort required for its resolution.
- Bug mitigation not supported. There are not policies or guidelines for avoiding bugs. All activity is address to solve bugs once they have been produced.
- Workflow bottlenecks, produced when a developer receives more assignations of bugs than he/she can process due to the lack of control over the load of the developers.

It has been reported numerous times that one of the most problematic limitations that SMEs must overcome to develop services built upon OSS solutions is how rapidly they identify effective ways for interacting with the main development community that supports these projects. Broadly speaking, we can see that OSS development teams should have devoted a significant amount of resources to meet these missing requirements. In reality, we see that most of these teams cannot face these challenges, mainly because of the lack of time (that must be devoted sometimes to figure out the next development steps to be performed) and the loosely-coupled nature of the interrelationships between community members.

## 2.4 Target Market

Target Market is the specific group of customers that a company aims to capture. They have been identified as people with needs or wants that can be met with the products or services from this company. It is important to know your target market and what they want and need, in order to produce a successful product/service.

ALERT, by producing a solution for the better open source development process, will have a strong impact on the increasing use of OSS. It is expected that ALERT will have a very high impact in three specific groups of interest:

1. Open Source communities both international and national, such as KDE, Apache, Debian, OW2, Morfeo... These communities must face complex management problems to distribute tasks among their contributors. Small and

large projects are contributed by many developers in these communities and are not always synchronized in solving a bug, due mainly to the lack of awareness about what other developers are doing. Although this is the more focused target market for ALERT, the ALERT results can be easily applied to other markets such as the software industry market. We assume that this market is not a traditional market where the product/service is sold for a price as a unique way of business. The ALERT code will be delivered under Open Source model and different exploitation ways are suitable. Thus the communities will be able to download the ALERT results from our web site and be free to use and contribute it under the rules of the established licenses. In the case of this market, direct revenues for ALERT partners in monetary terms are not expected. However the use of ALERT results by OS communities will disseminate these results and it may affect indirect business, by providing advance services to third parties (companies involved in OS communities, freelancers, public administration, etc). There exists also an untouchable benefit in respect to this market that is the reputation of ALERT project and its partners in the OS world: to become a reference project.

2. Large companies and SMEs developing OSS. As we have observed in the market research, enterprises are developing OSS more and more. In the same way as OSS communities, the management of developers is complex and is missing effective management tools for these kinds of projects. Enterprises even have more pressure as most of these solutions are running on production level infrastructures with demanding requirements in terms of high availability, robustness and security constraints. This market does imply the procurement of monetary profits to project partners. The customization, development of advance features, configuration and installation of ALERT system are services that ALERT partners may offer to these companies.

3. Research Community. In spite of the fact that evident project results are the software components and that these ones are the most valuable results for OS communities and software industry, other results are also produced by the project such as the idea, the methodological approach, the conceptual architecture, the ontology, the pilots, etc. This kind of results is mainly valuable for the scientific community that can see in them the basis for future research or complementary work to their current research. This market is composed of IT researchers mainly focused on events processing, semantics, information extraction or notification management. Monetary revenues are not expected from the exploitation of this market, but again reputation and recognition will be the alternative profits.

# 3 ALERT Results

The ALERT system aims at improving the coordination between open source developers by maintaining awareness of community activities through real-time, personalized and context-aware notification.

ALERT goal is to create an active collaboration platform (i.e., a virtual actor that interacts with the developers) that aims at identifying different interactions, suggesting actions and bringing past interactions into the developers' attention enabling developers to work better together.

For that, ALERT extracts information from different sources from developers' environments (mail, forums, chat, bug reports, code repository, etc.) using a set of sensors installed in these community tools. Then, it analyses the collected information extracting metadata, annotating the data and storing it using an ontology. Different events are triggered according to the actions and interactions happened in the community. These events can be a notification, a recommendation or the execution of a service.

The implementation of this project idea will provide the following type of results:

- Conceptual approach
- Use cases
- ALERT ontology
- Software components
- Pilots
- Scientist publications

The ALERT **conceptual approach** is represented by the project idea and the conceptual architecture conceived for describing the proposed solution. The project idea rose from the perception in the OS communities that the management of the developers' coordination could be improved. A better process in bug resolution definitively would contribute to this improvement. Thus, the innovation is not only to find the way to provide an answer to the communities' problem but also to find the right technologies to do it in a best way. The challenge to combine semantics, event-driven processing and information extraction techniques to implement the solution proposed becomes a result itself. It can be a proof of concept for the implementation of future solutions to similar problems in other contexts. The conceptual architecture described in D1.2 Conceptual Architecture deliverable shows a picture about how to build the solution and which components are required for that.

The ALERT **use cases** have been modelled using an adaptation of S-Cube methodology. They can be found in D1.1 As-Is To Be Analysis and in the S-Cube repository. The use cases model the behaviour of the ALERT users (developers, customers, gatekeeper, leader and administrator) in the ALERT system. They have been described using the template proposed by S-Cube project and additional use case diagrams have been developed. This result is useful to check the feasibility of S-Cube methodology as one more example of application of the methodology.

The ALERT **ontology** will model the conceptual dependencies between community, content and interactions. Ontologies will be used as a backbone for defining the

conceptual architecture of the collaboration platform. At building the ALERT ontology, existing ontologies have been reused (such as QualiPSO, DOAP, FOAF, etc) and it is expected that further reuse of our ontology will be done in the future by others modelling the same knowledge.

The ALERT **pilots** are three based on the three OS communities represented in the project: KDE, OW2 and Morfeo. Three communities have experienced room for improvement in the bug resolution process and the ALERT solution can offer them a way to increase the efficiency in the code evolution and maintenance. The D1.1 As Is and To-Be Analysis deliverable describes the three communities and how the current bug resolution workflow is. On the other hand, the D8.1 Evaluation Plan shows how the trials of ALERT technology will be carried out and which projects are selected for deploying the software implementation. The conclusions and recommendations that will be produced after the trials execution will be an interesting result that can be reused from the ALERT pilots. This information will be described in D8.3 Evaluation Report due at M30.

Several **scientist publications** will be produced along the project mainly by research partners. These publications will be a probe of the research work achieved and can be an inspiration for further work to others or even to ourselves. The publications are also the means of dissemination in journals, web site or events.

The **software components** are described separately in the next section as the most relevant and complex result to describe and deliver.

## 3.1 Software components

In order to implement the desired functionality, ALERT is developing the following set of components:

- Sensors that will be used to monitor all information sources relevant for the bug resolution and notify about a change
- Knowledge extractor from structured sources (KESI) that will extract information from the structured sources
- Knowledge extractor from unstructured sources (KEUI) that will extract information from all unstructured data
- Metadata service that will create and store the semantically-enriched information and provide query and reasoning facilities
- Profiler (STARDOM) that will identify developers and create/update their profiles based on their activities
- Interaction pattern editor that will enable developers to describe situations relevant for notification
- CEP engine that will discover interesting situations
- Action service that will execute actions related to the detected situations
- Search service that will enable users to specify their information needs and provide support for detecting different relationships between the bug reports
- Recommendation service that will suggest a most competent developer to resolve an issue and a set of issues that are most suited to the developer's competence

- Visualization service that will provide different visual perspectives on the provided results
- The ALERT web UI that will be used for the interaction with the end-users.

The below picture shows all the components and the interactions among them:

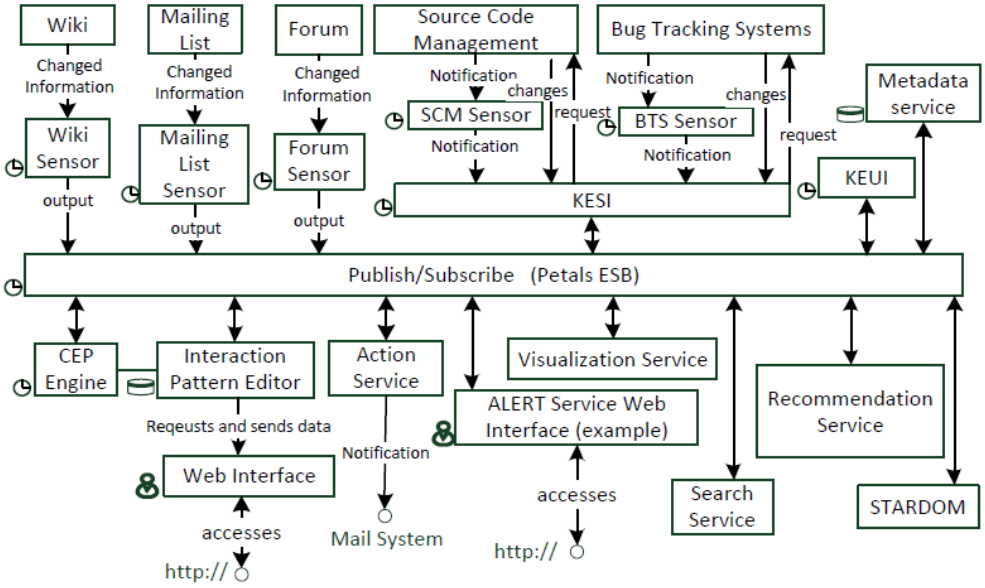


**Figure 5 ALERT components interactions**

The complete description of components and the conceptual architecture description can be found in D1.2 Conceptual Architecture deliverable.

Despite the detailed technical description of the ALERT components is in the above mentioned deliverable, next table summarizes a brief description of each one; which is the problem every component solves; the technical innovation provided; and the added value brought to the project. This information will be useful for identifying potential market and exploitation strategies for each component.

| Component | Description | Problem solved | Technical Innovation | Added Value |
|---|---|---|---|---|
| **Knowledge extractor for structured information (KESI)** | Extraction of knowledge from structured data sources (SCM, BTS) | Acquisition and linking of data extracted from structured sources in local system | First tool of this type supporting simultaneous data extraction from different structured sources | Easier interpretation of data and context extracted from structured sources |
| **Knowledge extractor for unstructured information (KEUI)** | KEUI will be responsible for semantically enriching all textual posts (mails, bug reports, forum posts, etc.) from all monitored data sources. It will also store the posts in order to support information retrieval. | Identifying the semantic meaning of the posted textual data. Storing posts in a form appropriate for efficient information retrieval. Assisting the search service in providing results to use-cases. | KEUI will provide innovative ways for annotating informal text. It will also implement several novel approaches to solving the use-cases, such as finding bug duplicates. | KEUI will play a very important part since it will process unstructured text which is one of the main sources of information in ALERT. |
| **Metadata Service** | Metadata service will provide support for creation, retrieval, reasoning about and storage of ontology-based metadata. Additionally, it will trigger semantically-enriched event-based communication. | The service will be the ALERT-specific ontology API that will be used by all other components/services. In this way, other components/services do not need to take care about management of knowledge (i.e. creation, integration, reasoning, usage, sharing, storage, etc.), as the metadata service will provide such | Metadata service creates metadata objects by using object factory class. Its purpose is to instantiate, interlink, preserve and to provide access to knowledge base. Instead of working with the ontology triplets, with the help of the metadata service all ALERT components/services will work on the level of ALERT entities such as bugs, users, methods, etc. | This service will be shared by all other components/services as it will provide an abstraction from the ontology triplets to the level of ALERT entities such as bugs, users, methods, etc. |

| Component | Description | Problem solved | Technical Innovation | Added Value |
|---|---|---|---|---|
| | | support. | | |
| **Profiler (STARDOM)** | STARDOM is responsible for creating and maintaining a developer's profile. More specifically, it focuses on enriching the profile with information about the expertise of the developer. The expertise of the developer will be modelled and stored in a competency model. | The modelling of the developer's expertise will assist in providing useful recommendations about which developer should fix a bug based on his profile. | STARDOM uses a novel combination of several metrics in order to calculate the Competency Index of a developer (expertise level). These metrics are: ITS Activity, Static analysis of source code, Number of issues fixed, Time since last commit, Number of commits that have caused an issue, Mailing list activity. Number of commits that fixed an issue, Time since last mailing list contribution, API usage count, Time since last ITS action, API introduced and SLOC. | The fact that STARDOM uses a wide variety of metrics to calculate the Competency Index results in a more accurate profile than profiles built by existing techniques. |
| **CEP Engine** | ETALIS is event processing framework which includes an expressive language for complex patterns, and an efficient execution model. The framework is logic-based, and feature deductive capabilities exploitable in event processing, as well as, in (re-)actions and workflows triggering. | Detection of complex events and reasoning over events and background knowledge (by means of logic rules) | Standard and novel event operators developed, possibilities to discover a desired state in the process of pattern detection, aggregation functions supported, etc. | Provision of the right information to right people/components in right situations |
| **Interaction** | The interaction pattern editor | Creation and maintenance | Structure, (statistical) data and | Supporting the domain |

| Component | Description | Problem solved | Technical Innovation | Added Value |
|---|---|---|---|---|
| **Pattern Editor** | enables the developers to describe situations relevant for notification. It is the implementation of a methodology for the management of complex interaction patterns | of the interaction patterns | usage driven evolution of the interaction patterns | expert in creating new patterns, increasing the reusability of existing patterns, refining patterns and enabling a semi-automatic pattern evolution |
| **Action service** | This service helps users to subscribe to the different notifications which will be provided by ALERT system | Creation, modification and deletion of subscription and execution of notification | Dynamic subscription as well as sending notification in two ways: (i) sending email or (ii) pushing the notification to user web client | Execution of the actions related to the detected situations |
| **Search service** | The search service offers support for simple queries and search capabilities exploring interconnections among knowledge elements retrieved from ITS, SCM and other data sources. For example, ALERT users will be able to explore possible duplicates for a given issue report, other issues sharing certain characteristics (same module, similar topic) and some connections between issues and related source code elements. | It is not straightforward for managers and developers working in a software project to have prompt access to cross-linked relationships between knowledge elements from the same or different data sources. A simple search API enables them to query extracted knowledge to facilitate their daily decision making process | The main innovation introduced by this service is the combination of multiple input parameters characterizing extracted knowledge from different data sources to identify possible connections and relationships among these elements. This includes the combination of information from structured fields with semantic relatedness (CSM) and other techniques that help the system characterize and relate knowledge elements to answer searches received from | This is the first attempt to merge information from different sources of knowledge into coherent indicators to explore links and connections among these knowledge elements. |

| Component | Description | Problem solved | Technical Innovation | Added Value |
|---|---|---|---|---|
| | | | managers and developers | |
| **Recommendation service** | The recommendation service will be used in order to suggest developers that are considered the most knowledgeable to resolve an issue. It will also suggest issues to developers according to their areas of expertise and competency indexes | The recommendation service will be used to reduce the burden of bug triage by humans due to the fact that based on the accurate profile of a developer (WP4), it will recommend the right people for resolving a bug. | Collaborative filtering and content-based recommenders will be examined. Enhancing the accuracy of the recommendation service system could exploit probabilistic models.<br>We will choose the most appropriate technology that fits into ALERT, in order to provide a state-of-the-art and innovative recommendation service. | Automatic bug triage and avoidance of cost and time loss. |
| **Visualization service** | Visualization of the results of the use-cases. | Helping the user to more easily gain understanding of the presented data | In the visualization service we will implement new and intuitive ways for summarizing the results of use-cases. | Visualization service will improve the usability of the ALERT system by allowing the users to be more efficient in their work. |
| **SCM sensor** | Synchronous extraction of information from SCM | No similar tools already available in the market | Extraction of information from SCM as it becomes available | Real-time tracking of SCM updates |
| **BTS sensor** | Synchronous extraction of information from BTS | No similar tools already available in the market | Extraction of information from BTS as it becomes available | Real-time tracking of BTS updates |
| **Wiki Sensor** | Synchronous extraction of information from Wiki | No similar tools already available in the market | Extraction of information from Wiki as it becomes available | Real-time tracking of wiki updates |

| Component | Description | Problem solved | Technical Innovation | Added Value |
|---|---|---|---|---|
| **Mailing List Sensor** | Synchronous extraction of information from Mailing List | No similar tools already available in the market | Extraction of information from Mailing List as it becomes available | Real-time tracking of Mailing List updates |
| **Forum Sensor** | Synchronous extraction of information from Forum | No similar tools already available in the market | Extraction of information from Forum as it becomes available | Real-time tracking of Forum updates |

**Table 5 Description of the ALERT software components**

ALERT proposes a solution fully modular for two major reasons:

- Avoiding limitations of each component's license in the ways of distribution, giving the freedom to each component (owner) to keep the type of desired license.
- Diversifying the exploitation paths, allowing users benefits from ALERT as an overall solution or component by component depending on their needs.

As consequence, the result of the project is not a fully integrated platform ready for being downloaded and installed, but a set of components that can work together following some instructions for installation and configuration, or separately integrated in compatible environments. Next section gives more details on the exploitation strategy.

# 4 Sustainability Plan

The consortium has been doing a brainstorming in order to ensure the ALERT sustainability after the end of the project. The sustainability is a key aspect to warrant the project success, and for this reason the consortium started working on this since the very beginning of the project.

Firstly, we identified the key results. As we have seen in the previous section, the key results of ALERT project are the software components released as Open Source Software. Being all Open Source, the exploitation strategy should be aligned and may be different than when is closed source.

Secondly, we recognized the key users that are mainly developers and contributors from Open Source communities or industrial companies. Taking into account that the developers will consume ALERT results, we have investigated what resources these people are used to. The Open Source Communities is the point of contact between our key targets. OS communities are open ecosystems in which individuals interact and collaborate to solve problems or to exchange ideas. As our key target is used to work in communities, the ALERT consortium has decided to **create a community around ALERT**. It will be a toolset repository, where all components developed in the project will be available to download, including future releases of the software components, support and maintenance. In addition, the web site will be improved and it will include forums with related hot topics to discuss, articles, documentation, publications, blog, news, etc. to create an active community around ALERT.

The consortium believes that creating a community around the project is the best way to promote the technology transfer as the consortium offers directly the results to the interested users. It will be a connection between the project partners and the target market, the Open Source communities, large companies and SMEs and the Scientific Community.

There are twofold advantages by creating this portal:

- users will have a place to obtain useful information, download the components, find the right partner for support and maintenance... etc.
- and the project partners will sponsor an specific site where their own components are shown, a new marketing channel to announce their organization and services, a new channel to obtain new clients...

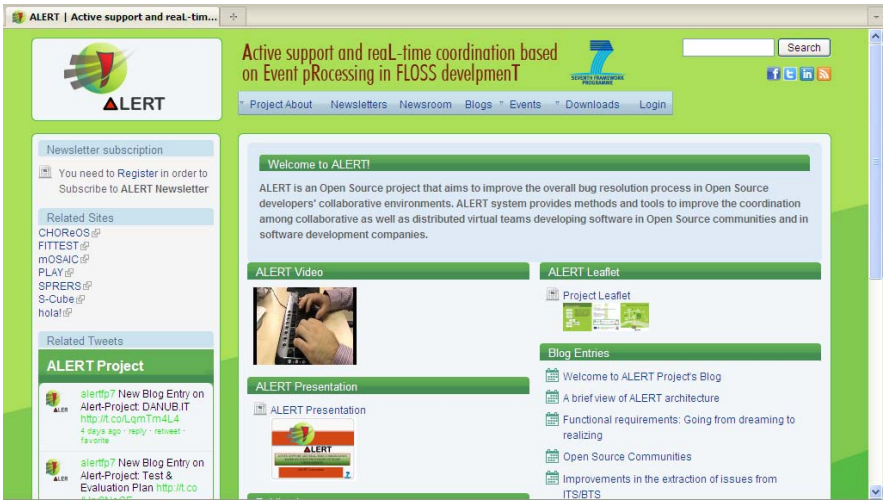


**Figure 6 Current ALERT website**

There are different phases to follow to develop the ALERT sustainability plan:

1. Define ALERT delivery
2. Identify ALERT business model
3. Define the community strategy.

## 4.1 ALERT delivery

The first phase is to define how ALERT results will be delivered to the users.

Regarding the software components, the potential ALERT exploitation paths identified at that moment were:

1. Release a bundle with overall project packaged and distributed together
2. Release components individually
3. Software as a Service (SaaS)

Let's see the analysis of the advantages and disadvantages of each option:

1. **Bundle with overall project packaged and distributed together:**

*Advantages*:

- User-friendly as it is easy to install
- More control on the evolution of ALERT results as must be distributed under GPL (restriction caused by some GPL components in ALERT)

*Disadvantages*:

- Do not cover all exploitation opportunities for the project partners, since some components are licensed under GPL. This would force some partners to release their own code under GPL as well, due to the copyleft clauses. Hence, this could pose a conflict with existing exploitation plans of partners

releasing their code under permissive licenses (e.g. to explicitly allow integration with third-party proprietary code)

2. **Release components individually:** Download them separately from a repository including a builder to locally build the overall functionality.

*Advantages*:

- Each component holds the license of the partner owner. Not viral effect to other components.
- Foster modularity of the system. The components can evolve separately; there are not limitations between components and organizations.
- More flexible for the future development of components

*Disadvantages*:

- Required a user with technical skills
- More complex installation

3. **Software as a Service (SaaS):** This approach implies the existence of a framework that allows the discovery, deployment, execution and monitoring of a set of services. The ALERT components have been developed as web services so their functionality is well defined, encapsulated and feasible to be deployed in a service oriented environment. Despite of it, the components are not ready as they are to be deployed directly in a service infrastructure. They would need some extra effort to allow their direct consumption by users following a SaaS model.

*Advantages*:

- User-friendly
- Users do not have to install the software as it is offered as a service
- There are no conflicts between licenses as are offered as a service

*Disadvantages*:

- Extra resources needed to prepare the components to be deployed in a service infrastructure.
- It is not considered explicitly in the Description of Work. According to the decision of the consortium and the exploitation strategy adopted, an allocation of efforts could be needed to achieve the required modifications.

The first option has been discarded at the moment due to the incompatibility between some viral licenses with the exploitation strategy of some partners. In case of relicensing of some of the components, this alternative could be reconsidered again.

The third option is to offer the results as a SaaS. This is an interesting option and it could be considered, but under the consideration that it requires an extra investment of effort since it was not considered initially at the proposal conception.

At the moment and after analysing all the possible exploitation paths, the consortium decided that the most appropriate option for ALERT results is the second one: **to release components individually.** This is the most feasible option in the project since it does not cause the incompatibility or viral effects between licenses, it is flexible

enough for the future development of the components and enables the exploitation strategies of all the partners. This option enables and maximizes the future sustainability of the project results. However, an analysis of dependencies among components is required since some of them should be released together, because they need each other to work. In these cases, the licenses compatibility must be ensured and clear indications for the users have to be provided.

Of course, this choice is submitted to evolution of the components, the conclusions of further market analysis and changes in licensing of components or partners' strategy.

The delivery of the attached results of the project such as the Conceptual approach, Use cases, ALERT ontology Pilots and Scientist publications will be delivered through the portal.

## 4.2 Business Model

### 4.2.1 Introduction

There are many definitions of what a business model is. According to Chesbrough in its book "Open Business Models" [5], the business model is a useful framework to link ideas and technologies to economic outcomes. At its heart, a business model performs two important functions: value creation and value capture. First, it defines a series of activities that will yield a new product or service in such a way that there is net value created throughout the various activities. Second, it captures value from a portion of those activities for the firm developing the model.

ALERT has already defined the value creation in section 3. The value capture can be done by developing different activities that captures the revenue.

### 4.2.2 Generic OSS Business models

There are a number of open source business models. According to Chesbrough, these models include [5]:

- Selling installation, service, and support with the software
- Versioning the software, with a free version as an entry-level offering and other, more advanced versions as value-added offerings
- Integrating the software with other parts of the customer's IT infrastructure
- Providing proprietary complements to open source software ( these increase in value as the cost of the open source code falls; one version of this strategy is to create a creative commons and then build a proprietary products or services on top of the commons) .

Another typology of OSS business models is done by a European project called Qualipso [8] that provides a state of the art on the OSS business models used in the industry. These are described below:

1) Independent software vendor business models

a. Distributing copies of an Open Source product for a fee

b. Proprietary offerings based on an Open Source kernel

c. Dual licensing

d. Franchise

e. Delayed availability of the successive releases

f. Legal issues rose

2) Ecosystem enabler business models

a. Network of companies

3) Software as a service vendor business models

a. Subscription model

b. Proprietary Software as a Service based on Open Source

4) Service provider business models

a. Providing support around the open source software

b. Providing maintenance of the developed code

c. Integration with other open or close source systems

d. Training and certification about the software

e. Software customization for a specific use or environment

f. OS components' selection

g. Process certification

h. Risk assessment and insurance

5) Combination of Open Source components with hardware

6) Brand selling

7) Advertisement

8) Global fee redistributed to authors

In addition, as Richard Stallman stated (founder of the Free Software Foundation) “There is no contradiction between selling copies and free software. In fact, the freedom to sell copies is crucial... Selling them is an important way to raise funds for free software development”.

### 4.2.3 The ALERT Business Model

The need for software do not finish once you get the product. Once the product is made available, a number of related needs arise, such as consulting, installation, configuration, maintenance, support and training, for which certain customers are willing to pay.

Through the ALERT portal, there are a number of results offered for free, such as the software components, articles, publications... and others offered as paid based services, such as integration, consulting, support, maintenance, training... The interested organization may find the right partner to buy the services required. The portal, with interesting articles and contributors in the field, acts as bait for the additional services offered by the partners.

There are two different business models, on one side there are the project partners or industrial companies advertised in the community, and on the another side, there are the business models of the community itself in order to guarantee the sustainability of the community. The figure below summarizes the different business models.

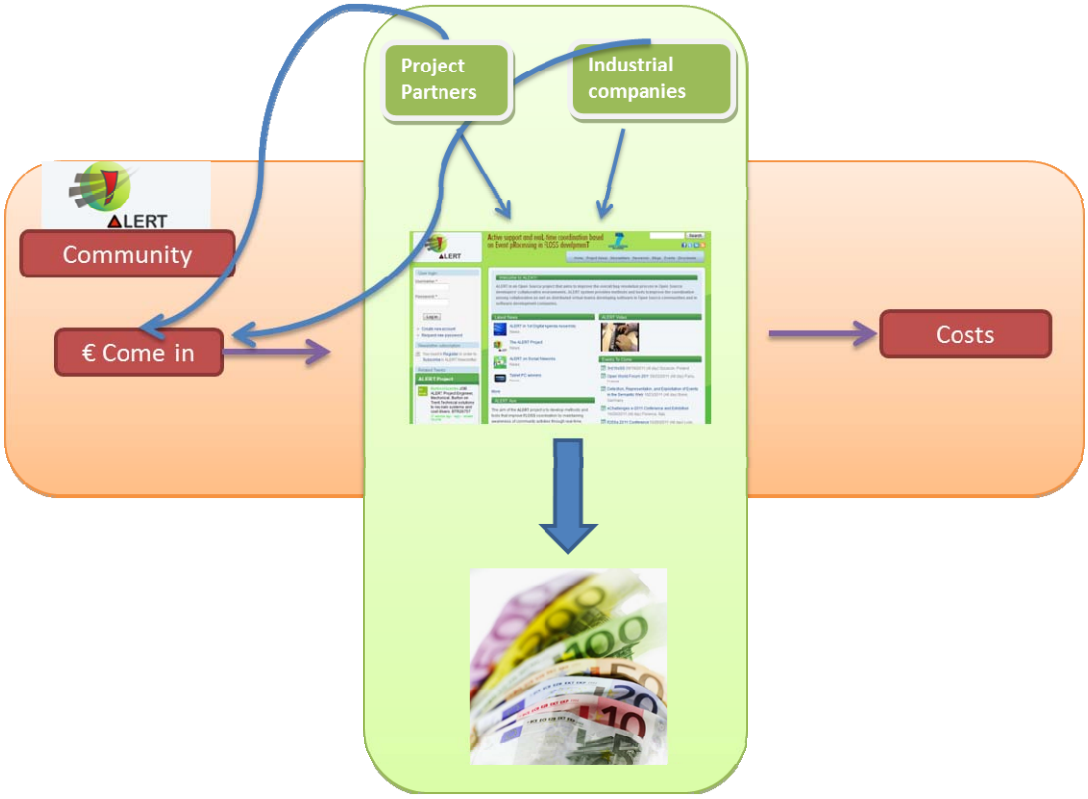


**Figure 7 ALERT Community business models**

The project partners and industrial companies active in the community will reap the benefits by capturing clients from the community. Being in the ALERT community gives to those companies more chances of getting a large client base. In addition, being a member of the community will make easier paths for collaboration.

The sustainability of the portal itself will be in charge of the project partners advertised in it and offering its services, paying a fee for advertising or offering infrastructural support to the portal. In addition, external advertising will be considered to be included, in order to assure the sustainability of the portal.

### 4.2.4 Community strategy

OS communities are open ecosystems in which individuals interact and collaborate to solve problems or to exchange ideas. Consist of individuals, who contribute to build, write, an application (FLOSS development) and voluntarily do activities further than the realms of software development.

We believe that the most appropriate type of community for ALERT is the Special Interest Groups (SIGs) as it is aimed for research and learning technologies.

One of the most important aspects in a community is the communication. Communication requires feedback as well as sender, receiver and a message. The communication in the ALERT community will be asynchronous, which is based on

store-and-forward principle, both persons involved in communication do not have to be online at the same time.

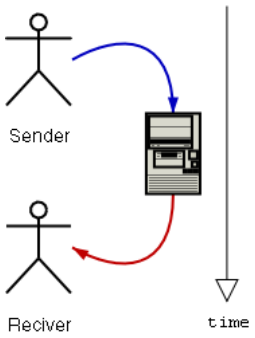


**Figure 8 Asynchronous communication**

Common communication tools are e-mail, interactive web pages, such as forums, articles, news... In ALERT the communication will be also through the tools developed in ALERT which will be the demonstrators of the effectiveness of the tools itself.

The **main goal** of the community is to reach the sustainability of ALERT results and their evolution beyond the project life.

There are several tasks to be undertaken to create a community, for the moment we identified the following:

1. Market research: identify and analyse relevant companies/organizations which may be interested to be involved in the community. This is to ensure that there is a market to create the community. ALERT needs collaborators to contribute, providing content, offering in-kind services or even to sponsor the community. Firstly, it will be the project partners and gradually external organizations will be involved, too.
2. Content provider: since the very beginning all the ALERT results will be available on the community, the components to download, the public deliverables, the publications, the uses cases... etc. Afterwards, all this content will be updated and realised new ones, through the inclusion of new contributors and collaborators.
3. Paid advertising: investigate options for paid advertisements of ALERT using Google Awards and other, similar, companies.
4. Link building: link building refers to asking other sites to link to ALERT – taking into account that we would usually be expected to link them first. It is a good chance to promote the website to other sites and get visitors through these.
5. Branding: Branding is the process involved in creating a unique name and image for a product in the consumers' mind. Design and create an ALERT brand.
6. Social Networking: nowadays the social networks results vital to promote anything, and so the existence of ALERT community will be done accordingly.
7. Monitoring: Monitor the growth of the community.

The tasks will be distributed among the partners taking into account the effort available, the core competencies of the partners and their role within the project. The tasks will be analysed during the next period of the project.

### 4.2.5 Next steps

The next steps regarding sustainability are:

- Confirm within the whole consortium and the European Commission that the community building is the best option for the sustainability of ALERT
- Synchronize efforts between exploitation and dissemination tasks
- Identify and analyse the tasks needed to develop a community
- Organize and distribute the tasks between the project partners
- Improve the website in order to build a community
- Working on all the tasks

The option of building a community around the project results is not an easy way to proceed. There are many tasks to carry out and many people should be involved. Creating a community entails higher risk, but also higher impact when succeeds. The ALERT consortium is devoted to create a community and we are looking for interested people outside the consortium to join us and test, use and even contribute to our developments.

# 5 Initial SWOT analysis

SWOT analysis is a strategic planning method used to evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in a project or in a business venture. It involves specifying the objective of the business venture or project and identifying the internal and external factors that are favourable and unfavourable to achieve that objective.

- **Strengths:** characteristics of the business or team that give it an advantage over others in the industry.
- **Weaknesses:** are characteristics that place the firm at a disadvantage relative to others.
- **Opportunities:** external chances to make greater sales or profits in the environment.
- **Threats:** external elements in the environment that could cause trouble for the business.

Identification of SWOTs is essential because subsequent steps in the process of planning for achievement of the selected objective may be derived from the SWOTs. As ALERT is a project which aim is to develop new software products (services), this analysis is done to identify the internal and external factors related to ALERT results.

| Strengths | Weaknesses |
|---|---|
| • Important OSS communities in the Consortium<br>• Real test cases and datasets entail project outcomes tailored to community/developers needs<br>• Low adoption barriers thanks to good integration with already existing development tools and processes.<br>• Unique combination of input data and context from different sources (SCM, BTS, mailing lists, etc.) in a single assessment solution.<br>• Improving effort management in software development environments with strict demands.<br>• Increase project/community performance reducing the complexity of time-consuming tasks (e.g. duplicate issue detection).<br>• Generation of business and new services on top of ALERT outcomes | • Resulting algorithms might need fine customization to effectively address specific projects/communities.<br>• Scalability issues may arise in very large projects/communities, depending on the size of datasets to be processed.<br>• Performance will be dependent on available resources of existing IT infrastructure.<br>• Complexity inherent to the combination of knowledge extracted from different data sources.<br>• Deployment in target infrastructures may need specific customization, since it is difficult to automate the complete installation process.<br>• Imbalance in data sources can affect accuracy of assessment, especially regarding new code or developers.<br>• Particular traits of projects and their datasets may pose a challenge for the accuracy of assessment results |

| | |
|---|---|
| • Large and growing market | |
| **Opportunities** | **Threats** |
| • Lack of similar technological solutions in current OSS development scenarios<br>• Increasing awareness of cost in an OSS development<br>• Increasing business support to OSS<br>• Emerging Technologies | • Defending OSS Use in Mission-Critical Scenarios<br>• Other organizations release a similar product.<br>• Lack of success in community building |

**Table 6 SWOT analysis**

# 6 Preliminary IPR and Licensing analysis

The licensing of the project results has been a concern for project partners from the beginning of the project. Due to the previous experiences and expertise of the consortium, the analysis of the licenses of the components or documents to be produced was identified as an essential task in the context of the exploitation work package.

At the time of licensing the ALERT results, we have considered in a different way the documentation and the software. All the project documentation has been delivered under Creative Commons licenses:

- Attribution-no derivatives (BY-ND) for dissemination material. The material can be redistributed but no derivatives works are allowed.
- Attribution-Share alike (BY-SA) for the public deliverables (all except the management reports and exploitation plans). The deliverables can be redistributed even for commercial purposes and you can build upon them as long as they credit you and the licenses keeps as the original one.

In the case of software results, the licensing issue has required a deeper analysis and work explained further in this section.

There are several reasons why the consortium thinks it was mandatory to deal with the licensing aspects from the very beginning of the project:

- Licensing of project results (software, documents, other material) affects the exploitation strategy. Although the licensing has several technical aspects, the type of licensing affects the exploitation paths to be adopted and that is the reason why the licensing analysis has been done in the framework of the exploitation plan.
- It is necessary to ensure the compatibility of licenses of different project components
    - Those ones that will be developed during the project
    - Those ones that are reused from previous projects
- The analysis of the licensing status must be done at the beginning of the project, in advance to the development. In this way, we avoid working on incompatible components or at least we know where the incompatibility is and we assume the implication in the exploitation strategy.

This preliminary analysis can be slightly modified along the project life.

In order to proceed with the analysis of the software components licenses and the impact on project exploitation, we have defined a working plan described below in several steps:

1) We set up a concrete group of people in the consortium to work on this issue. We selected from the partners those persons with more expertise in licensing. One representative per partner with knowledge in licenses management and with the technical knowledge about the components was incorporated to this group.

2) We defined a table with information about software components and their used licenses. According to the project architecture (conceptual picture), we identified the ALERT software components to be developed and all the dependencies with other modules or libraries they have. For each of them, all the licenses involved, the owner and the link to the license were completed. The filled table can be seen in 9.

3) Then, the nominated group analysed the overall table of components and licenses and evaluated the implications in the exploitation by:
   a. Reading the different licenses terms
   b. Consultation to FLOSS legal experts when required
   c. Periodic audio conferences for discussing

   The analysis included the study of the compatibility among the different licenses of the components to be developed and with their respective dependent components. Not all the components have defined already their license, but at the moment the summary of the adopted licenses are:

   - 6 components under GPL v2
   - 4 components under LGPL v3
   - 2 components under BSD

   Some conclusions of the analysis are:

   - The components under GPL license will make the overall distribution of ALERT solution as GPL. The detection of this conflict causes that the bundle distribution of ALERT will not be possible since the adoption of GPL license does not cover the exploitation intentions of all the partners.
   - In the dependencies of every component, some LGPL v3 components are using Apache v2 and we have detected a possible conflict between Apache v2 and LGPL v3.

4) Afterwards, the overall consortium had a brainstorming to discuss the delivery strategy in accordance to the analysis done by the restricted group about the licensing. The conclusions are described in section 4.1

# 7 Preliminary individual exploitation plans

This section collects all the individual exploitation plans for the ALERT partners. They have different interests on what ALERT will produce and due to their respective nature (industry, academic, research or community) their business goal is completely different and the benefits they expect from the project as well.

We have classified the exploitation plans according to this nature and the requested information is slightly different. These plans can evolve during the project and an updated version will be provided at the end of the project.

## 7.1 Industrial Partners

### 7.1.1 Cim Grupa Doo (CIM)

*7.1.1.1 Partner profile*

CIM is a software and consulting company with rich experience in business and technology consulting and large scale IT integration background.

CIM is regarded as the top institution in Western Balkan (particularly in the countries of Former Yugoslavia) for software research & development, business process management, quality management consulting, and publishing. Besides Serbia, CIM has satisfied customers all over the region. The same is true for the joint venture partnerships, which are existent Europe-wide.

Ever since 1991, when CIM was founded, we invested significant resources in helping organizations from the region of Southeast Europe improve their business processes, by providing software support and consulting services. Over time, we were able to create a knowledge base, build-up know-how, and gain experience in process management.

In order to be at the cutting edge of business process improvement using software development technologies, CIM developed its own process-oriented platform for component software development - OPISys™. OPISys™ stands for Object Process Integration System, designed for information systems development, based on processes with component modules. OPISys™ operates in Intranet/Internet environment and provides crucial support for effective and efficient business performance. OPISys™ information systems are customer-tailored solutions, enabling flexibility at the business process design layer, workflow execution layer, and quality monitoring layer. Moreover, 15+ software solutions for quality tools and methods developed by CIM are widely implemented in the leading organizations and R&D projects enabling the ultimate benefits of automated quality management.

Given the ALERT business model, CIM will provide value adding services around the ALERT exploitable results. CIM, as a highly active software vendor, will firstly exploit the ALERT infrastructure internally, with an aim of improving in-house software development process. Additionally, CIM intends to achieve benefits from the know-how planned to be used/developed in the project, with a special accent placed on the issues regarding semantic technologies and event-driven architecture.

Considering the fact that the service component is becoming increasingly important in the OSS adoption process, CIM will provide a wide array of services that will support the highly diverse needs of its users during different phases (i.e., installation, deployment, operation and maintenance).

#### *7.1.1.2 Identification of ALERT business/transfer opportunities*

In the light of the ALERT project's main goal of improving the software engineering process in commercial software developing companies and FLOSS by developing a platform, embedded in an open source software engineering environment that will enable software companies to provide high quality open source software and professional software services, thus enhancing the competitiveness of the software industry, CIM, as a software developing and consulting company will exploit ALERT project results with regard to its relevant business activities. CIM's exploitation plans are mainly related to the external exploitation of ALERT components and services via:

1. *Improving software development process*

As a software development company, CIM plans to exploit the ALERT platform in the context of more agile bug resolution process. Apart from incorporating the ALERT solution in its own software development life cycle, CIM will promote the ALERT solution on seminars addressed primarily to the IT decision makers, thus preparing the soil for ALERT results exploitation in other software development systems.

2. *Custom Developments*

Whenever a solution is required which gets out of the scope of the projects goal, a customer might require a customized solution in which a piece or the entire software has to be integrated into another solution, has to exchange data or communicate with specific hardware or software. Though the overall ALERT solution fits the requirements it would be very hard for the customers' own IT department to integrate it to the requirements specified. At this point the "learning" cycle would be less cost effective as compared to the usage of programmers already familiar with the ALERT platform and underlying code. For this reason programmers may be leased to these customers on hourly / monthly leasing fee or by contract. CIM would provide consultancy with regard to implementing the customized solution and most effective use of this new tool, including technical support and other software-related services requested by the user. In this way, the knowledge gained in ALERT project will be put into practice via customized solution and consultancy provided for its most effective implementation.

3. *Support Services*

Installation support to any technical installation means physical presence of technicians or specially trained supporting personnel on site. Also, comprehensive manuals should allow customers to undertake installations on their own, though the expertise of a service provider can and will speed up the process of integrating a platform of the ALERT size and expertise into specific environments.

4. *Training*

CIM will provide training to customers for acquiring the necessary knowledge to operate the platform effectively and productively in any development environment. Educators with broad knowledge in software development will visit customers sites and provide seminars to the end-users and if required to the supporting IT staff.

#### *7.1.1.3 Potential Addressable Market and Customers*

CIM mainly searches for exploitation opportunities in the Republic of Serbia, but also within other countries in the Western Balkans. As being an active participant in software R&D projects partially sponsored by the European Commission within FP6 and FP7, CIM will build on the results of ALERT project and identify other research

projects in the field of software development as potential users of the ALERT results. Therefore, the potential market for exploitation may be expanded to the stakeholders within the whole Europe and its open source community.
Furthermore, CIM recognised software development companies and members of FLOSS community as the main potential markets for exploitation of ALERT solution. As being itself a software developing company, CIM will first exploit the results of this project internally in order to optimize its current software management practices. This will provide us with first-hand experience with ALERT solution, which will hopefully show results in promoting the solution further in the open source tool developing community and software development companies. CIM will share the idea of ALERT project on seminars which address primarily to IT community in Serbia and Western Balkans, and offer it as an innovative solution for enhancing the development performances, improving the quality while simultaneously cutting the costs of the software development processes.

#### *7.1.1.4 Customers' Needs and Expectations*

The main target audience of the ALERT platform are software development companies and FLOSS community, which need to improve their flexibility by shortening the time needed for software development and enabling better responsiveness of their developers and efficiency in information management, decision-making and productivity.

Since bug resolution is the integral part of software development and probably the most time-consuming activity in the process, the potential customers of ALERT open source solution are anticipated to expect the ALERT system to be a facilitator in bug resolution and producing the quality software via faster and more efficient channels of interaction.

#### *7.1.1.5 Value Proposition*

CIM will use the results of ALERT project to introduce the software development organizations and FLOSS to new concepts and technologies. Apart from the state-of-the-art solution they will be offered, the ALERT target customers will be offered full consulting support and appropriate trainings as add-on services in the framework of an existing contract and without charge, for the purpose of the most effective exploitation of the gained knowledge and introduced innovative solution.

ALERT solution will bring benefits for software development companies and integrators of FLOSS in reducing the time and human effort needed and providing a new software technology for improving scalability and predictability of distributed systems by providing a service that will help FLOSS developers in their daily tasks.

The capacity of ALERT to track individual events within the software development process and to generate knowledge that can be returned to developers and community members is critical to foster scalability and sustainability of these initiatives. Improving the efficiency of FLOSS developers, automating important and time-consuming tasks like detection of duplicated bugs, and focusing developers effort towards those pending issues best fitted to their skills and previous expertise, are also key factors to enhance the predictability of these projects. Eliminating those factors that may act as time drains for the development team will help these projects to focus on more productive activities, taking up development of FLOSS to new levels of performance never achieved before.

In this context, eliminating the hot spots that waste time and effort could counteract these downsides, leaving enough room and energy for the team to concentrate on providing a more dedicated service to the end-users, clients and other collaboration partners. ALERT will contribute to mitigate these problems, offering precise and frequently updated information about the specific needs that must be addressed in the development process.

### 7.1.2 Atos Sociedad Anónima Española (ATOS)

#### *7.1.2.1 Partner profile*

Atos is an international information technology services company, delivering hi-tech transactional services, consulting, systems integration and managed services. Atos is focused on business technology that powers progress and helps organizations to create their firm of the future. It is the Worldwide Information Technology Partner for the Olympic Games and is quoted on the Paris Eurolist Market. Atos operates under the brands Atos, Atos Consulting & Technology Services, Atos Worldline and Atos WorldGrid. Atos annual revenues are of €8,7 billion, and it has over 78,500 business technologists worldwide in 42 countries.

Atos Research & Innovation (ARI), node of Atos R&D&I located in Spain & Turkey is a worldwide reference for innovation for the whole Atos group. ARI performs projects usually within an international scope that combine advanced technological developments with economic exploitation. ARI work is focused on several R&D areas: public sector, health, industry, energy & utilities, Telecom, IT and media. ARI plays the role of source of innovative ideas to be used by Atos sales and technical staff.

The Open Source Software use and development has become a strategic line in the company since some years ago. Many solutions and applications have been built under the principle of including open standards and open software. The gain of competitiveness and the increment of contributions to open source communities are major objectives behind this strategy.

Atos is member of Morfeo community. As a major IT player in Europe, the contribution of Atos towards this community is mainly the promotion of this community among its customers and the possibility to exploit some software hosted in those communities thanks to the inclusion of it in some of the integrated solutions that Atos may do for its customers

#### *7.1.2.2 Identification of ALERT business/transfer opportunities*

Atos is one of the biggest IT players in Europe. The company develops millions of code lines daily all over the world in large projects formed by geographically distributed teams working in collaborative environments. For such amount of software, the efficient management of it in all the phases is crucial for ensuring the profits in every project. The company usually uses common code repositories and collaborative tools for the communication and interaction of their developers. Even in numerous cases, the Atos developers must collaborate with external developers to the company, using also common collaborative environments.

This context envisages the need of ALERT results for the improvement of Atos development process. Basically, the identified business opportunity for Atos respect to ALERT is the possibility to install ALERT components in our common collaborative

environments in order to improve the resolution of bugs that arise from testing processes applied to our code and monitoring by our ticketing systems.

The ideal situation would be to install the overall ALERT system, but it will depend on:

- Maturity of components after the project end. Not all the components could be at the same level of maturity at the end of the project.
- License of each component. We could not install any component that would be incompatible with our code licensing
- Feasibility of integration with our corporative systems

In case the overall system installation will not be feasible, an analysis of what component is more relevant for us will be performed

#### *7.1.2.3 Potential Addressable Market and Customers*

Atos covers multiple markets such as banking, financial services, insurance, media, oil and industry, public sector, retail, telecom, transport, utilities and health. As a major software integrator, Atos develop every day millions of code lines within the different sectors.

Atos is using OSS tools to perform its own developments. The know-how gained in building such environments can be used to build similar platforms for customers. Atos also provide support on the most used OSS. As an example Atos is supporting JBoss for the French Ministry of Finance.

Specific division within Atos is the Offer & Development division, which mission is to build the company's overall software offer.

Our main customer for ALERT results is Atos itself. The adoption of ALERT results in our own development environments will be the main benefit of the project for us. Of course our final customers will experiment the improvement in our delivery of code caused by the use of ALERT components.

Additionally, we could also offer additional services from ALERT results such as the installation and configuration of ALERT system or the components thanks to our expertise in one of the ALERT pilots.

#### *7.1.2.4 Customers' Needs and Expectations*

Atos and Atos customers' needs in relation to support in OSS development is:

- reducing time and effort for OSS development
- effective ways for interacting with the main development community
- enable user-visibility
- provide effective interactions between teams
- avoid duplication of bug reports

#### *7.1.2.5 Value Proposition*

As said before, our main customer for ALERT results is Atos itself. Clearly, the final customers of our solutions will benefit of our improvement in bug resolution by offering a better maintenance and evolution of software in less time and saving money.

### 7.1.3 EBM Websourcing SAS (EBM)

*7.1.3.1 Partner profile*

EBM WebSourcing is an IT open-source editor (SME) established on 2004 in Toulouse, France. With more than 30 people specialized in middleware technology and service oriented architecture, EBM WebSourcing provides a strong added value in term of service infrastructure technologies as well as modelling and development tools.

EBM WebSourcing is working on two main families of open-source products that allow building cloud computing platforms:

- Petals platform family: Petals ESB (http://petals.ow2.org) and Petals orchestration engine (BPEL).
- Petals modelling family: Petals Master (Dragon, http://dragon.ow2.org), Petals web modeler (Web-based BPMN editor) and Petals wave (6Napse Web 2.0 collaborative platform).

EBM WebSourcing is founding member of the OW2 consortium, it has a sit at the board and leads the technical council. EBM is also associated member of the OASIS standardization body, member of the NESSI ETP and founding member of the Interop-VLAB virtual laboratory.

EBM has an extensive know-how on managing complex open-source projects and relative communities. In particular EBM has been driven the Petals ESB communities since 5 years. Moreover EBM will take advantage of its involvement into the OW2 consortium for attracting other open-source communities into ALERT.

Besides this, EBM participates in many collaborative research projects funded by the European Commission (Soa4All, SYNERGY, GENESIS) or the French government

*7.1.3.2 Identification of ALERT business/transfer opportunities*

Bug duplicate identification

Consolidated view of all information relative to a bug, including bug tracker description, wiki page, and forum discussion.

*7.1.3.3 Potential Addressable Market and Customers*

No market or customers addressable, as this is an internal tool to make our development easier.

*7.1.3.4 Customers' Needs and Expectations*

Not targeted to a customer, but to community users and developer. See OW2 users' needs and expectations

*7.1.3.5 Value Proposition*

Not targeted to a customer, but to community users and developer. See OW2 value proposition.

### 7.1.4 Corvinno Technologia Transzfer Kozpont Nonprofit Kozhasznu Kft (CORVINNO)

*7.1.4.1 Partner profile*

Corvinno Technology Transfer Center Nonprofit Public Ltd. is a spin-off company born at the Corvinus University of Budapest. The aim was to establish an organization that completely focuses on the exploitation of the gathered knowledge and experience. Corvinno team has demonstrated its aptitude and efficiency in leading organizations from the public and the private sector.

The common feature of Corvinno team possesses a unique combination of innovation, management and technical expertise and above all sharing the same professional and ethical values. In accordance with the innovation life cycle, Corvinno is proud of leading prestigious research projects in the field of ICT, disseminating its results through conferences, publications and trainings, and offering consultancy and solutions to its clients

*7.1.4.2 Identification of ALERT business/transfer opportunities*

Corvinno will use the project results to extend its consultancy portfolio. Exploitation strategy will be oriented toward selling "services" around the system, either in the combination with editing tools or by "manual" modelling. We expect interest especially towards solutions capturing knowledge from unstructured sources and capturing interaction based knowledge

*7.1.4.3 Potential Addressable Market and Customers*

Corvinno is running consultancy projects at Hungarian banking and telecommunication companies, where large software development and IT service departments are active. Results from ALERT are well suitable in these environments

*7.1.4.4 Customers' Needs and Expectations*

Our clients are especially keen on cost efficiency and return on investment on complex IT systems. Anything that can help them achieving better/quicker results by using less resources – especially human resources – may raise their interest

*7.1.4.5 Value Proposition*

ALERT can help clients to minimise efforts, especially in the application incident/problem/change management processes. Using the application as a knowledge management tool can decrease required time to close related issues, thereby increasing business value of the appropriate systems.

## 7.2 Academic Partners

### 7.2.1 Forschungszentrum Informatik An Der Universitaet Karlsruhe (FZI)

*7.2.1.1 Partner profile*

FZI is a non-profit research and technology transfer center comprising 14 R&D teams – each of them directed by a Professor also holding a chair for Computer Science, Electrical or Mechanical Engineering, or Business Administration at the University of

Karlsruhe – Germany's oldest and one of its most successful Technical Universities. FZI helps its partners and customers in applying novel information technologies for realizing new and better products, services, and business processes. FZI is a member of the World Wide Web Consortium (W3C), member of NESSI, it participates in the IBM Center for Advanced Studies (CAS) programme, it established together with Microsoft a .NETbased center for Innovative Software Concepts, and it won several contracts from the European HPMT Programme to act as a Marie-Curie Training Center.

FZI was founded in 1985, currently employs ca. 120 researchers plus additional student assistants, and had a turnover of ca. 11 MEURO in 2008. It has outstanding experience in scientific research and industrial development projects in regional, national and international cooperations, as well as in providing technology consulting services, for instance feasibility studies, technology scouting, market studies, prototype development, etc. Through its multidisciplinary, close-to-the-University approach and its close collaboration in manifold networks (with its University sister institutes, several spin-off companies, partners in European Networks of Excellence, etc) it can ensure on one hand that newest methods are applied and further developed and the highest level of ICT research excellence can be reached, and on the other hand that research results can widely be spread into the scientific community.

FZI is represented in the ALERT Consortium through its IPE department (Information Process Engineering). IPE employs methods and tools from Computer Science and Economics for the analysis, assessment and design of intelligent information logistics in organization-internal and cross-organizational business processes, in cooperations, value networks, and virtual organizations. The iCEP team that has been involved in the ALERT project operates within IPE as an interdisciplinary unit engaged in research and consulting activities, which lie in the area of complex event processing

#### *7.2.1.2 Identification of ALERT business/transfer opportunities*

As a general remark which can be concluded from the organization form as a non-profit technology transfer center it should be mentioned that FZI – in order not to lose its tax-free status and in order to fulfil the mission it was created for – is not allowed to make significant earnings from selling products or services. FZI's success is not an extremely good financial balance sheet coming from many sold software licences, but FZI's success is the success of its partners and customers. So, commercial projects serve the purposes to fill the funding gap created in only partially financed research projects (like ICT projects), to better show practicality and application benefits of fielded solutions based on our technology, and earn some financial reserves to allow people finishing their PhD without project pressure and/or allow us to develop new, promising technological topics before we can acquire projects about these topics.

But, adhering to the FZI mission, besides such commercial contracts, exploitation through new research projects, increased technology outreach and visibility through dissemination and education activities, successful idea transfer to project partners and spin-off companies, and even knowledge transfer through people who do their PhD thesis at FZI and transfer their knowledge to the industry or other research contexts, is a success for us.

Having said this, the following exploitation opportunities for FZI have been arisen from the ALERT project:

- **Follow-up research projects -** The most obvious exploitation opportunity for a research institute is to define follow-up research projects which take up and further develop (deeper technical sophistication or consolidation or broader application) ALERT topics, results, and application ideas, or continue work with ALERT partners. Concretely, ALERT opened four exploitation threads of this kind for FZI: further elaborate infrastructure for event-driven systems, further work on interaction pattern management, work on event-based reasoning and pruning or further work in the software engineering sector.
- **Consulting services -** With the knowledge that FZI built up during the ALERT project, consulting services, prototype development, and demonstrator projects can be acquired from industrial partners. Within ALERT, this concerns again two threads of work: on one hand, management of events and patterns, and on the other hand, application of the event-driven systems.
- **Teaching content -** Not typically understood as a "direct" exploitation asset, the creation of dense and provenly useful training material on event driven systems, complex-event processing, ontology engineering, etc. has nevertheless an "indirect" exploitation value for FZI: since the public funding of FZI by the federal state of Baden-Württemberg is also based on the expectation that FZI contributes to state-of-the-art teaching and education to Baden-Württemberg students, especially of the University of Karlsruhe, it is useful for us to have access to worked-out teaching material about up-to-date, scientifically valid, but practice-oriented topics. Having noticed that event processing are a major future technology area, and that real-time awareness is one of the very important trends in IT nowadays, it makes sense to use ALERT tutorials as input for teaching. Here, not only the well elaborated ALERT tutorials can be used in lectures, but also parts of the software infrastructure, for practical exercises.
- **Personnel transfer -** The last area of subtle, but not neglectable, impact stemming from research projects is that of transferring educated personnel from a research institution to other research institutions or to industry partners. Typically, this happens after successful completion of a PhD thesis

#### *7.2.1.3 Potential Addressable Market*

A growing general interest in CEP, together with the existence of a number of proven, convincing real-world application scenarios will massively increase the prospective consulting, teaching, and development coaching business of FZI, with customers from big (software) industry and SMEs in the different areas, and from big industries with their own software departments. Specific ALERT results can be productized by spinoff companies. Having CEP technologies as an acknowledged basic technology for real-time applications will also improve our general visibility and credibility, since we can look back at a long track record in this area, since the beginning of the movement.

- Market for the CEP components:
  - University students
  - Research community
  - Software industry
  - Big industry and SMEs acting in different domains (traffic management, logistics, eHealth, smart grids, sensor networks, social media monitoring, etc.)
- Market for the ALERT system:
  - FLOSS development communities

- Large industries and SMEs developing FLOSS
- Software companies with distributed software development

*7.2.1.4 User Needs and Expectations*

There are different types of users with the different needs and expectations:

- **Students**: A unique opportunity to learn about the latest developments and open challenges in an emerging area such as real-time stream processing. This will be done through:
  - Lectures on fundamental and applied aspects of the complex-event processing and event-driven software development with hands-on tutorials and experimentation, providing the participants with a solid understanding and experience to further advance research in this field;
  - Organization of summer schools (e.g. sensations, https://sites.google.com/a/senzations.net/www/) which will enable the acquisition of new skills and expertise while learning about recent advances and important open research challenges and directions;
- **Research community**: Possibility to advance research on this real-time stream processing from both theoretical and practical points of view.
  - Excellent research solutions related to real-time processing: identification and resolution of the research challenges and some practical experiences in developing and using such methods;
  - Organisation of the scientific workshops (e.g. SWESE2011 at ISWC 2011) which will lead to further exchange between practitioners and researchers working on issues related to complex event processing by providing a forum for discussing the major challenges of the area and the different approaches being taken to resolve them;
  - Organization of the competition for event-driven applications (e.g. DEBS 2011 Event Processing Challenge http://debs2011.fzi.de/index.php/challenge) by creating awareness of the potential of the event-processing paradigm for developing various, challenging real-world applications;
- **Industrial partners:** Transfer of knowledge into the customers' companies:
  - by performing market and technology scouting, coach brainstorming for innovative technology usage, creating awareness through workshops, demonstrator, prototypes, etc. – thus helping the industrial partners with a faster and better understood and more effective uptake of new technologies;
  - by supporting the partners in finding the right decisions: about how to create, store, distribute, retrieve, integrate, etc. real-time information – altogether, the efficient exploitation of events as a productive resource and as a value-creating product, e.g. through customized services, prototype development, and demonstrator projects that can be offer to industrial partners;
  - by providing deployment services of the ALERT platform (installation of the software or customized deployments) in software organizations in order to train these on how to use the platform;
- **FLOSS users:** Provision of better information management in relation to issue resolution, making it easier to manage issues and target developer resources towards fixing them.
  - Leaders: More efficient management of open source software projects by better traceability of the development tasks and the corrective actions brought in

response to issues and by improving productivity and ensuring the quality of developments;
- Volunteer developers: a better collaborative experience with the project by improving the communication and cooperation on bug identification, fixing and testing between the contributors;
- End users: Improvement of their satisfaction by shortening the response time to correct failures

*7.2.1.5 Value Proposition*

**Follow-up research projects** – In general, the value of new research, development, and transfer projects for a non-profit technology-transfer institution is self-evident. So, we give here a clearer picture of some (not all) concrete projects: At FZI, we can differentiate between two kinds of projects as ALERT follow-ups:

- projects exploiting the ALERT approach for complex event pattern management, e.g. ARtSENSE - Augumented RealiTy Supported adaptive and personalized Experience in a museum based oN processing real-time Sensor Events, http://www.artsense.eu/
- projects exploiting application of complex-event processing technologies, e.g. ReFLEX - Reinforcing FLEXibility of SMEs by dynamic business process management, http://www.reflexforsmes.eu/

The value creation is self-evident: defining and elaborating new research trends and bringing them closer to practical application, is the core business of FZI. From the collaboration with other partners, new ideas arise, quality is increased, resources are increased, and impact can be bigger. ALERT provides a new paradigm for management of complex situations and many individual technical solutions which can be further consolidated and further developed.

Here, the direct beneficiary is obviously FZI, plus the respective project consortia who win some complementary public funding for their research work. But also, in the long-term, the open source community, because funded research shall, of course, produce generally useful results and new technologies and methods which can be leveraged for a wider take-up in software industry and open source community. Obviously, continuous involvement in current research trends is a crucial success factor for FZI and also creates opportunities (in terms of content, contacts, and visibility) for more commercial projects.

**Consulting services –** Basically, FZI's core business in the commercial area is selling high-tech consulting, to big industries and SMEs, especially in the IT sector; so we help early technology adopters to be faster and scientifically better accompanied in new technologies and new usages of technology; in this context, the following concrete activities can be listed as ALERT-based:

- "Viral selling strategy" to SMEs: from time to time, SMEs issue contracts to FZI for helping them in the implementation of "down-to-earth" technology, such as B2B solutions or SOA infrastructures. Normally, such organizations are difficult to convince that they should invest in high-tech (and, maybe, high-risk) approaches. Hence it is difficult to sell them ideas like the ALERT approach, or even "total solutions" following a new paradigm. So, the best approach for such customers is to deliver to them a basic solution using state-of-practice, consolidated techniques, but adds some little, conservative extension, delivering

some not expected added value, maybe not even asked for. In this way, such defensive companies can become acquainted with new technologies. This is an approach; FZI constantly tries to apply with its conservative customers;

- Exploitation to/through spin-off companies: Long-term goal for FZI is to continuously sell technology consulting to spin-off companies (as a direct source of income) and (as an indirect source of income) invest in a strategic partnership with spin-off companies to support them in their customer projects with respect to, e.g., semantic event processing;

**Teaching content** - FZI continuously contributes to teaching at the University of Karlsruhe, as well as other public education institutions in the federal state of Baden-Württemberg. It is foreseen to continue the teaching activities in these areas.

Besides these non-commercial teaching activities, FZI together with the University of Karlsruhe and Ontoprise GmbH launched the "Semantic Web Akademie", a commercial Semantic Web education offering addressing German-speaking decision makers on one hand and technicians interested in hands-on experience on the other hand.
The general aspects of value creation for FZI through teaching have already been described above in the description of the exploitation opportunity. The specific facets of ALERT comprise: the well worked out, extensive set of teaching material can be prepared as real-time event processing tutorials and the extensive description of ALERT use cases. The specific opportunity is to gain potential commercial customers through seminars and teaching activities. A mid-term achievement can be to acquire from an open teaching event publicly announced, a specific bilateral contract for an in-house workshop.

**Personnel transfer –** Transferring well educated and motivated people from the ALERT project team to other research institutions and industries is an important multiplier for results, approach, philosophy, etc. There are no direct revenues, on the contrary. However, the network of "event-infected" people grows, and so do the opportunities for follow-up activities in international collaborations. Hence personnel transfer is a significant means to create a sustained impact.

### 7.2.2 Universidad Rey Juan Carlos (URJC)

*7.2.2.1 Partner profile*

GSyC/LibreSoft is part of the Telematics and Computing Department (GSyC) at the Telecommunication Engineering School of Rey Juan Carlos University. Libresoft was founded as a consequence of the long interest in libre software, and the strong relationship with the libre software community.

The group is focused on research and development related to libre software. Its main line of research is the quantitative study of libre software development, specifically from an empirical software engineering point of view.

Some of the European-funded research projects from this area, in which GSyC/LibreSoft is working or has been involved in the past, are FLOSSMetrics, which its aim is to provide a huge publicly available data set with data mined from libre software repositories for their further study; QualiPSo aimed to define and implement the technologies, processes and policies to facilitate the development and use of FLOSS software components, to develop innovative and reliable information systems; QualOSS, whose goal was tomeasure the quality of libre software projects, especially

regarding evolvability and robustness; and FLOSSInclude, that addressed the impact of libre software in several countries around the world.

*7.2.2.2 Identification of ALERT business/transfer opportunities*

The ALERT outcomes that offer potential business or transfer opportunities for Libresoft are mainly the software components developed and improved to support the assessment process, including:

- The KESI and KEUI modules for acquisition and preparation of structured and unstructured data from libre software projects.
- Other services supporting the assessment process, such as the search, recommendation and visualization services.

On one hand, this software should be suitable to support further empirical analysis of data sources from libre software projects, which is one of the main research lines of our group. On the other hand, it also provides an opportunity to offer independent consulting services to third-party stakeholders interested in either analysing their software projects or integrating these components in their own IT infrastructure

*7.2.2.3 Potential Addressable Market*

As a result of the business and transfer opportunities presented above, 2 markets could be addressed:

- The research market of tools to support empirical analysis of libre software projects (historical data or in real-time) which is an emerging sector with multiple applications.
- Consulting services build on top of the assessment process that can be implemented using ALERT. This is also an emerging market with several notable examples in the private sector, such as Black Duck

*7.2.2.4 User Needs and Expectations*

The perceived user expectations are different depending on which market we consider:

- In the research market of empirical software engineering there exist little or no support for synchronous analysis of information retrieved from software projects as it is generated. This would open a whole new set of possibilities for researchers in this field to further develop suitable methods for assessing and improving this process proactively (instead of reactively, as it is forced today due to the limitations of current available solutions).
- As for consulting services, there exist a long-standing need for reliable and open tools that merge information from different sources of the software project to assess the managers, improve the task distribution among project members and reduce the fraction of effort devoted to very time-consuming tasks such as detection of possible duplicated bugs, inspection of relationships between issues and code or recommendation services that facilitate the daily job of developers.

*7.2.2.5 Value Proposition*

As for the exploitation interests of Libresoft, the main value proposition of ALERT is twofold:

- Enable the synchronous analysis of information gathered from structured and unstructured data sources in libre software projects, something beyond the scope of current alternative solutions in this field that only support analysis of historical data logs.
- Combination of inputs from different sources and context enrichment (by means of the ALERT ontology), enabling a more comprehensive description of the situation and evolution of the software project, and more precise assessment

### 7.2.3 Institut Jozef Stefan (IJS)

*7.2.3.1 Partner profile*

Jožef Stefan Institute (IJS) is the main research institution for natural sciences in Slovenia. It consists of over 800 researchers within 25 departments working in the areas of computer science, physics, chemistry and biology. The Artificial Intelligence Department is a research group working in the areas of machine learning and data mining. It has approx. 40 researchers covering different aspects of data analysis with special emphasis on textual data, social networks/graphs, complex data visualization and cross modal analysis.

In the recent years the research shifted towards semantic technologies, where the main goal is to combine modern statistical data analytic techniques with more traditional logic based knowledge representations and reasoning techniques. The department developed several software tools that will be used in ALERT, among others: Text-Garden suite of text mining tools, OntoGen system for ontology learning, Document-Atlas for complex visualization

*7.2.3.2 Identification of ALERT business/transfer opportunities*

The opportunities from ALERT that could be exploited by JSI are the following:

- Consulting services. Although ALERT system will be designed to support as general projects as possible it is likely that some components of the system will have to be fine-tuned for different projects. JSI could suggest to companies how to use ALERT in order to achieve the optimal performance.
- Spin-of companies. JSI is a non-profit research organization therefore products that are created are typically monetized through spin-of companies created in association with JSI. In case of ALERT, a new company could be created in order to provide code customizations for different customers. Since processing of unstructured text is one of the core parts of ALERT it is likely, for example, that in order to provide high-quality processing of non-English text, a new set of pre-processing tools would have to be developed.
- Follow-up projects. For JSI, ALERT is one of the first projects in cooperation with open source communities and we would be happy if it would become one of many

*7.2.3.3 Potential Addressable Market*

The market that could be addressed by this result are the software companies that would benefit from using the ALERT system and would require either consulting services or an adaptation of the solution (such as support for a different language).

*7.2.3.4 Customers' Needs and Expectations*

The main need of the users would be to have a reliable solution that would seamlessly integrate various information sources used in the project. They expect that the system would be responsive even for large projects and that it would be successful even when processing text that is not written in English language

*7.2.3.5 Value Proposition*

From the consulting perspective we are offering to the users a high quality service that would enable them to achieve the best performance using the ALERT system.

For users with additional requirements we can offer any necessary modifications that would be required by them. Examples of such modifications could include a custom built Annotation ontology, lemmatisation for a custom language, pre-processing of text for a custom language, implementation of stack-trace parsers for a non-supported programming language, etc

### 7.2.4 Institute Of Communication And Computer Systems (ICCS)

*7.2.4.1 Partner profile*

The Institute of Communication and Computer Systems (ICCS) is a non-profit private law body associated with the School of Electrical and Computer Engineering (ECE) of the National Technical University of Athens (NTUA). ICCS was established in 1989 by the Ministry of Education of Greece in order to promote research and development activity in all diverse aspects of computer and telecommunications systems and their applications. The active research personnel of ICCS comprises of 80 faculty members of ECE/NTUA, 25 senior researchers and more than 150 researchers.

In ALERT the Information Management Unit (IMU) research unit of ICCS participates. The Information Management Unit is a multi-disciplinary Unit engaged in research and development activities in Information Technology Management. The mission of IMU is to enable the development of knowledge-driven organizations and its main focus it to design, develop and validate innovative models, methods and tools that leverage the creation, sharing and use of information and knowledge at the individual, team and organizational levels. The staff of IMU includes three faculty members, seven post-doctoral senior researchers and twelve researchers. Since its establishment IMU has actively contributed in more than thirty research and development projects.

*7.2.4.2 Identification of ALERT business/transfer opportunities*

Since ICCS is a research institute the main exploitation channels for ALERT results will be through new research projects, through increased visibility via dissemination and education activities, as well as through the potential creation of spin-off companies. Hence we expect the following exploitation opportunities from the ALERT project:

- research projects: we will aim to define and take up further research activities in the areas of our work in ALERT. The main potential avenues for further research are hybrid recommender systems; and software developer expertise.
- educational material: due to the close connection of ICCS to the School of Electrical and Computer Engineering of the university, we will aim to further enhance existing, or develop new, graduate and post-graduate courses, by including material and ideas from the ALERT project.
- spin-off companies: members of the IMU unit have already been involved in setting-up three spin-off companies mainly by exploiting the results of research projects. Whether this will also be the case with ALERT results is still too early to say. Nevertheless there is a potential business opportunity for the recommendation algorithms developed in the project

*7.2.4.3 Potential Addressable Market*

The potential market for the ALERT components of ICCS besides the open source community are software companies (especially large ones) as well as big industrial companies with their own IT and software development departments. Of course for these markets we would have to implement (minor) changes in our approach since the closed corporate environment is not the same as an open source community. Moreover the potential market of our expected results includes the research community, the FLOSS development communities, as well as industries and SMEs developing FLOSS

*7.2.4.4 User Needs and Expectations*

We consider different user types who have varying expectations:
- FLOSS developers and industries and SMEs developing FLOSS expect consistent recommendations for addressing the bug triage issue;
- the research community expects improved algorithms for the assignment of developers to bugs

*7.2.4.5 Value Proposition*

The value proposition of the research outcome of ICCS in ALERT is focused on:

- an innovative competence/expertise index for software developers that takes into account many different metrics coming from structured and unstructured data;
- a novel approach for recommending software developers to bugs based on their overall activity and expertise level

## 7.3 Open Source Communities

### 7.3.1 K Desktop Environment EV - KDE

*7.3.1.1 Community profile*

KDE is a community of users and contributors that creates a set of workspaces (desktop interfaces) for Linux and a set of libraries and applications useable across all major computing platforms. There are about 800 active contributors and millions of users in home, educational and business situations. KDE's software ranges from a complete office suite to painting applications and enterprise email and calendaring. The software

is available in over 55 languages and a new version is released roughly every six months, with monthly maintenance releases.

#### *7.3.1.2 Identification of ALERT transfer opportunities*

ALERT should offer a number of results useful to KDE:

- Better issue tracking, particularly by identification of duplicate issue reports and therefore making the list of issues more manageable and useful
- Greater awareness and interaction across teams in KDE by automatically finding related issues that affect other teams and suggesting community members with relevant knowledge
- Automated warnings on emerging issues, such as new code methods implicated in multiple issue reports
- Better community metrics, keeping track of progress and in software and development of teams, giving improved metrics to measure our community and to present to potential partners such as sponsors to demonstrate the value of our work

#### *7.3.1.3 User Needs and Expectations*

The users include software developers and users. The users want to find it easier to make relevant reports of issues in the software and to feel that these issues are being dealt with. Better statistical reports on activity can aid this. Developers want to minimize time spent interacting with issue reporting systems and therefore need timely and relevant information on the most urgent issues, most common issues and how issues relate to each other. Early warning of problematic areas of code can assist developers in spotting problems early before building multiple layers of problems.

To be accepted, the ALERT system should however be non-intrusive as far as possible, requiring little compulsory change to existing workflows. Therefore a layered approach in which few changes are forced on the user, but more information is available on demand is highly desirable

#### *7.3.1.4 Value Proposition*

The main advantage of the ALERT system should be two-fold. First it should save developers time and free up time for creating better software. Secondly, it should make it much easier to gather relevant statistics on community activities and performance over time. The target users of KDE should experience ALERT mainly through better KDE software being achieved. As a secondary benefit, users should find the reporting of software issues more intuitive, with better guidance and suggestion of relevant answers to the required fields in the issue tracker. For example, ALERT may suggest the relevant software components automatically and identify if an issue has already been reported.

### 7.3.2 Morfeo

#### *7.3.2.1 Community profile*

Morfeo can be considered a community of communities specialized in platform SW developments. It is oriented to Spanish and Latin American contributors.

The community it's focused on acting on two main areas: as an incubator of R+D+i and as an experimentation field of the win-win model. This implies that their objectives involved different aspects such as making visible new research areas, enforcing collaboration of universities' groups, adopting an Open Source software development model and contributing to develop Open Source standards related to service-oriented architectures

*7.3.2.2 Identification of ALERT transfer opportunities*

All the ALERT results are suitable to be interesting for the Morfeo community.

As many members of Morfeo community are researchers, the conceptual idea, the posed solution and the produced publications are relevant for them.

For the industrial members of Morfeo, the experience in the pilots testing and validating ALERT could be also very fruitful.

And of course for the administrators of the community and the developers involved in Morfeo projects, the ALERT software components will be very valuable for improving the management in solving bugs and optimizing the time they contribute to the community projects.

*7.3.2.3 User Needs and Expectations*

The bug resolution in Morfeo follows a well-defined workflow, but still some weaknesses have been identified.

The Morfeo community users need and expect that ALERT will:

- Facilitate the identification of duplicated bugs avoiding that more than one developer will be working solving the same bug.
- Decrease bottlenecks allowing a user to know better whom to assign a bug.
- Improve the search methods used to find similar bugs inside the community. It is really useful to access to similar fixed bugs, although they are not reported in the same project.
- Improve the integration between the BTS and other communication tools

*7.3.2.4 Value Proposition*

The installation of ALERT components in the Morfeo community infrastructure will impact in a larger efficiency in the resolution of bugs detected in the projects.

The administrators of the community infrastructure will offer all the projects in the forge the possibility to adopt the ALERT system. They will provide the installation and configuration of ALERT components to those interested projects or will grant the project administrator to do it.

A previous dissemination campaign will be required for allowing community projects and developers are aware of the ALERT solution and benefits. Of course the evaluation results coming from the Morfeo pilot (based on LibreGEOSocial project) will be crucial for convincing other projects the goodness of ALERT software.

Further contributions to ALERT software from Morfeo community will be also fostered. Part of the success of the ALERT sustainability goes through the implication of OS communities in the components development (evolution and maintenance)

### 7.3.3 OW2

*7.3.3.1 Community profile*

OW2 is a business-oriented community to provide middleware software. Its members are companies, organizations, academics, and individual members. The main activity comes from company members, and OW2 helps to connect them together and to do business.

OW2 has almost no constraints regarding development. Code has to be open source licensed (with a validity checker) and middleware, that's all. OW2 gives some guidelines on code development, to keep more coherence in the consortium, and partners choose to follow them or not.

*7.3.3.2 Identification of ALERT transfer opportunities*

Bug duplicate identification.

Consolidated view of all information relative to a bug, including bug tracker description, wiki page, and forum discussion.

*7.3.3.3 User Needs and Expectations*

The users want their bug to be solved as fast as possible, and to get informed of the bug status, even if it is not planned to be corrected yet

*7.3.3.4 Value Proposition*

ALERT may help the user by spotting a duplicate bug he would have entered.

The global impact may be a better and clearer bug resolution process. Old pending bugs will be spotted easily, so developers can correct them, propose a workaround, or at least explain that it will not be solved yet.

# 8 References

[1] Economic and Social Impact of Software & Software-Based Services, D3 – Baseline Scenario for 2020, Pierre Audoin Consultants SAS (PAC), January 2010

[2] Economic and Social Impact of Software & Software-Based Services, D2 – The European Software Industry, Pierre Audoin Consultants SAS (PAC), July 2009

[3] Matthew Cheung, Alan Dayley, "Market Share Analysis: Operating System Software, Worldwide", Gartner, April 2010

[4] As-Is and To-BE Analysis, Deliverable 1.1 of ALERT project.

[5] Henry Chesbrough, "Open Business Models", Harvard Business School press, 2006

[6] Gartner Report, "Cool Vendors in Open-Source Software, 2011"

[7] Matt Aslett, Analyst, The 451 Group, "From Support Services to Software Services – The Evolution of Open Source Business Strategies", Open Source Business Conference in San Francisco, May 2010.

[8] Qualipso (Quality Platform for Open Source Software), IST-FP6-IP-034763, www.qualipso.org

[9] Gartner Report, "Hype Cycle for Open-Source Software, 2010"

[10] D1.1 As-Is To Be Analysis, ALERT consortium.

[11] D1.2 Conceptual Architecture, ALERT consortium

[12] D8.1 Evaluation Plan, ALERT consortium

[13] Vainio, N., Vadén, T., Oksanen, V., Seppänen, M. (2006). Elements of Open Source Community Sustainability. Helander N., Antikainen M. (eds.) Essays on OSS Practices and Sustainability. Tampere. (EBRC Research Reports 36)

[14] Estimating the number of active and stable FLOSS projects, Roberto Galoppini, 2007, http://robertogaloppini.net/2007/08/23/estimating-the-number-of-active-and-stable-floss-projects/ in FLOSSMETRICS project

# 9 Annex A: Preliminary IPR and Licensing analysis

Please find below a table with all ALERT components and their corresponding licenses.

| ALERT Component | WP | Description | Foreseen interface with other components | Exploitation ways | Component license | License text or URL | Copyright holder | Dependencies | License |
|---|---|---|---|---|---|---|---|---|---|
| Interaction pattern editor | WP5 | Designing and Evolution of Complex Event Patterns (in ALERTnamed as Interaction Patterns) | static link with CEP engine | as a service or packaged | LGPL v3 | http://www.gnu.org/licenses/lgpl.html | FZI | GWT | Apache 2.0 |
| | | | | | | | | SmartGWT | LGPLv3 |
| | | | | | | | | Java | BCL |
| | | | | | | | | Jena Semantic Web Framework | BSD |
| Action service | WP6 | enable user to subscribe to interesting ALERT notifications | called via web service | as a service; deployed in a platform | LGPL v3 | http://www.gnu.org/licenses/lgpl.html | FZI | java | BCL |
| | | | | | | | | maven | Apache |
| | | | | | | | | Apache CXF | Apache License 2.0 |
| CEP engine | WP5 | Detection of complex events and reasoning over events and background knowledge | The component listens to incoming events via a socket or a dedicated interface | It can be exploited either as a service or a standalone component | LGPL v3 | http://www.gnu.org/licenses/lgpl.html | FZI | SWI Prolog | LGPL |
| Search service | WP6 | Search capabilities through ALERT knowledge base | XML-RPC or similar API | Standalone or as a service | GPL V2+ | http://www.gnu.org/licenses/gpl-2.0.html | URJC | To be identified | To be identified |
| Metadata service | WP6 | The purpose of metadata service is to instantiate, interlink, preserve and to | called via web service | as a service; deployed in a platform | LGPL v3 | http://www.gnu.org/licenses/lgpl.html | CIM | Java | BCL |
| | | | | | | | | Apache | Apache License, Version 2.0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | provide access to knowledge base | | | | | | | |
| Knowledge extractor for unstructured information (KEUI) | WP3 | Extracts information from unstructured text | called via web service | as a service or packaged | BSD | http://www.opensource.org/licenses/bsd-license.php | JSI | OpenCyc | Apache 2.0 http://www.apache.org/licenses/LICENSE-2.0.html |
| | | | | | | | | Wordnet | WordNet 3.0 license http://wordnet.princeton.edu/wordnet/license/ |
| Knowledge extractor for structured information (KESI) | WP2 | Extracts information from structured text | | | GPL V2+ | http://www.gnu.org/licenses/gpl-2.0.html | URJC | CVSAnalY | GPL V2+ http://git.libresoft.es/bicho/tree/COPYING |
| | | | | | | | | Bicho | GPL V2+ http://git.libresoft.es/cvsanaly/tree/COPYING |
| Profiler (STARDOM) | WP4 | Creates a developer profile, by calculated competency indices per subject area, based on the activity of the developer | called via web service | as a service; deployed in a platform | GPL v2+ | http://www.gnu.org/licenses/gpl-2.0.html | ICCS | Spring Framework, http://www.springsource.org, 3.0.4.RELEASE | Apache License, Version 2.0. |
| | | | | | | | | SLF4J, http://www.slf4j.org/, 1.6.1 | MIT license. |
| | | | | | | | | Jackson JSON Mapper, http://jackson.codehaus.org, 1.8.1 | Apache License, Version 2.0. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Xstream, http://xstream.codehaus.org/, 1.3.1 | BSD license |
| | | | | | | | | Mysql JDBC Connector, http://dev.mysql.com/downloads/connector/j/, 5.1.16 | GPLv2 |
| | | | | | | | | Hibernate, http://www.hibernate.org, 3.5.1-Final | LGPL v2.1 |
| | | | | | | | | BoneCp, http://jolbox.com, 0.7.1.RELEASE | Apache License, Version 2.0. |
| | | | | | | | | Bushe Event Bus, http://eventbus.org/, 1,4 | Apache License, Version 2.0. |
| | | | | | | | | Junit, http://www.junit.org/, 4.8.1 | Junit Common Public License - v 1.0 |
| | | | | | | | | DBUnit, http://www.dbunit.org/, 2.4.8 | LGPL v2.1 |
| | | | | | | | | Joda Time, http://joda-time.sourceforge.net/, 1.6.2 | Apache License, Version 2.0. |
| | | | | | | | | SVNKit, http://svnkit.com/, 1.3.5 | TMate Open Source License. |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Commons Logging, http://commons.apache.org/logging/, 1.1.1 | Apache License, Version 2.0. |
| | | | | | | | | Commons Lang, http://commons.apache.org/lang/, 2,4 | Apache License, Version 2.0. |
| | | | | | | | | Commons Codec, http://commons.apache.org/codec/, 1,3 | Apache License, Version 2.0. |
| | | | | | | | | Freemarker, http://freemarker.sourceforge.net/, 2.3.18 | OSI Approved (http://freemarker.sourceforge.net/docs/app_license.html) |
| Recommendation service | WP4 | Recommends developers based on their expertis ( competency) | called via web service | as a service; deployed in a platform | GPL v2+ | http://www.gnu.org/licenses/gpl-2.0.html | ICCS | To be identified | To be identified |
| visualization service | WP3 | Visualizes the query information and presents the results to the user | called via web service | as a service or packaged | BSD | http://www.opensource.org/licenses/bsd-license.php | JSI | To be identified | To be identified |
| BTS sensor | WP2 | Live extraction of information from SCM systems | Probably REST interface (to be developed) | Standalone or as a service | GPL V2+ | http://www.gnu.org/licenses/gpl-2.0.html | URJC | To be identified | To be identified |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SCM sensor | WP2 | Live extraction of information from ITS sources | Probably REST interface (to be developed) | Standalone or as a service | GPL V2+ | http://www.gnu.org/licenses/gpl-2.0.html | URJC | To be identified | To be identified |
| Wiki Sensor | | Live extraction of information from wiki source | | | To be defined | | Not assigned yet | | |
| Mailing List Sensor | | Live extraction of information from mailing source | | | To be defined | | Not assigned yet | | |
| Forum Sensor | | Live extraction of information from forum source | | | To be defined | | Not assigned yet | | |

**Table 7 Licensing analysis**