Polaris4OS: a best practice for training and adoption of F/OSS in SME

Giulio Concas[1], Michele Marchesi[1], Andrea Piras[2], Stefano Sanna[2], Efisio Sciola[3],and Selene Uras[1]

1 DIEE, Dept. of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 - Cagliari, Italy
agile@diee.unica.it
WWW home page: http://agile.diee.unica.it/home.html
2 CRS4, Center for Advanced Studies, Research and Development in Sardinia
Polaris, Ed. 1, C.P. 25 - 09010 Pula (Cagliari), Italy
{piras, gerda}@crs4.it
WWW home page: http://www.crs4.it
3 ApritiSoftware!
Via Monteverdi, 12 – 09129 Cagliari, Italy
e.sciola@apritisoftware.it
WWW home page: http://www.apritisoftware.it

Abstract. The methodical adoption of F/OSS in SMEs raises critical problems related to issues such as technology, organization, culture and business model choices. In this paper, we describe Polaris4OS, a successful experience of F/OSS adoption among CEOs, managers and developers of ICT companies. The lack of F/OSS culture has been identified as the most critical obstacle for the exploitation of the new opportunities offered by the F/OSS communities.

1 Introduction

F/OSS adoption in SMEs introduces some non-trivial drawbacks. For business men and chief technology officer (CTO), F/OSS is critical since there are not case studies and deep analysis about it, making them lost in a huge collection of untrusted reports. On the other side, it is perceived by project managers as a critical resource, for which they have nobody to blame for misfunctioning or troubles. Finally, developers consider F/OSS as a synonymous of “Save as...” command in their browser: F/OSS is the cheapest way to access example code and get ready-to-run libraries and applications. Both business and technical people miss most potential opportunities

offered by F/OSS, like knowledge sharing, collaborative software improvement, large feedback basis, community support. We can consider F/OSS and its philosophy, spirit, methodology, tools and community, as an opportunity which requires skills and expertises outside the official learning path represented by universities, schools and certification centers. A new approach and learning program is needed to take advantages of the F/OSS ecosystem, as consumer and producer actor.
This paper mainly focuses on an "open source curriculum" defined and applied in the Polaris4OS project. It enables developers to effectively work in the F/OSS world and encourages chief executive officers (CEOs) to take advantage of these opportunities: it implies new learning methodologies, new skills to supersede closed and proprietary approaches, new business models to optimize company resources.

2 The Project

The main goal of the Polaris4OS project is to fill the gap between well established proprietary enterprise processes and the upcoming F/OSS model, based on collaboration and knowledge sharing. Eleven locally-based small and medium IT focused enterprises (SMEs) were involved in design and development of extension modules for existing open source frameworks. They had been selected according to the mission of each company and their market.
Polaris4OS consists of a preliminary set-up phase and three executive phases: the training based on a blended learning model, the collaborative design and the open source distributed development.

2.1 The Preliminary Phase

During the preliminary phase, companies were invited to evaluate existing state-of-the-art open source projects, and were asked to select projects to contribute to. They identified an Enterprise Resource Planning (ERP) and a Content Management System (CMS), which were considered as strategic solutions for their core business.
Then the project mentors met the companies to design and plan the next phases. Their core business had been totally based on proprietary software, with both custom solutions and off-the-shelf applications. Moreover, they were competitors in the same geographic area and they never cooperated. They had different experiences with a mixed background of Java and .NET environments, and each had its own software engineering methodology, usually based on practice and not on a formal approach. The preliminary meetings let to set-up a trusted environment in which companies agreed to share their experience, knowledge, resource, code and developers. The mentors identified the skills and the main competence areas needed to participate to selected projects. The key technologies were deemed to be object oriented programming, integrated development environments, collaborative utilities, versioning systems; the most important methodologies were considered to be design patterns, test driven development, continuous integration; the key processes were

deemed to be collaborative work and frequent iterations. The project effort was established in two man-days a week per company, for a timespan of nine months.

2.2 The Training

Since the initial developers skills were very different, we decided to adopt a flexible learning method would support a training process mixing classroom lesson and laboratory activities, on-line training and support, collaborative learning among learners, coaching and tutoring [1]. The learning model answers to our necessities is the blended learning model (BLM) [2] [3]. It was supported by the Moodle [4] learning management system (LMS) and by the collaborative knowledge base realized with the learning objects published by the teachers.
Our model consisted of three main learning moments:

- the collaborative learning, based on specific and practical activities during a frontal lesson;
- the self-learning phase. It gives to developers the possibility to use the material loaded in the LMS by teachers and to interact with coaches and tutors;
- the consolidation, when the developers came back in classroom to demonstrate the achievement of the administered concepts.

The BLM was realized in:

- 4 hours of classroom frontal lessons;
- 8 hours for individual study, collaboration, exercises based on the learning object prepared by teachers and having the support of tutors and coaches;
- 4 hours for classroom knowledge and capabilities assessment.

The LBS monitored developers activities, i.e. the time spent by each one on it and the result of on-line tests. From their analysis, we was able to define a training process tailored on individual necessities, creating a sort of learning customization, and to notice that the developers spontaneously create a community where people helped each other and shared materials and informations[5]. The community created in the first phase represent an important result for the project because it showed that the initial formers diffidence was overcame since the starting phase.

2.3 Designing

In such phase, we analyze the selected F/OSS projects and the definition of two specific application scenarios. To have the execution phases like a real project, we decided to have real customers therefore we selected the local administration of Pula, in Sardinia, and a private breedings company.
Meetings were taken to know customer needs and the software and hardware solution they used. It emerged the interest of the first customer for a tourist event alert system while the second one would like an automatic tracing system of animal growth for a pigs farm.
Tutors and coaches drove developers to evaluate the software solution and they found in Infoglue CMS [6] the most suitable F/OSS software to be extended

including the module required by the local administration and in Compiere [7] the ERP to extend and modify in way to implement the tracing system. Once selected the softwares, the developer team worked to define the tasks and a preliminary TO-DO list. Each item list was assigned to a developer.
Design activities were mainly conducted remotely by means of chats, web forum and mails, with weekly interactions for requirement definition, tasks identification and assignment.

2.4 Module Development

The developers implemented the extension modules driving their activities with concepts learned in the first phases and using the results of the design process. The development model was based on continuous integration and frequent releases in order to verify the correctness of the implemented software and the referential integrity of the required changes respect to the original databases structure.
During weekly meetings, the encountered problems had been analyzed and the possible solutions discussed by all developers for the adoption of the best choice. In such meetings, coaches and tutors applied a learning-by-doing model [8] in way to address the developers lacks raised in the course of this phase. A set of problems was related to the inexperience of the development F/OSS model like the remote and distributed collaboration model and the possibility of reusing the source code and the databases of the selected softwares.
For both of modules, the first development task was the installation of the original software and set-up of the integrated development environment (IDE). There were not available automatically procedures for the installation then the developer executed in separately steps the DBMS installation, the database creation, the Compiere or Infoglue installation and the connection between database and these softwares. Regarding to the IDE, the selected one was Eclipse [9].
The following tasks were tailored on the module. The alert system for Infoglue required the mapping of the new tables on the databases structure using Hibernate, the definition of the template for the web pages and the alert system. The system function realized were user registration, session and message management.

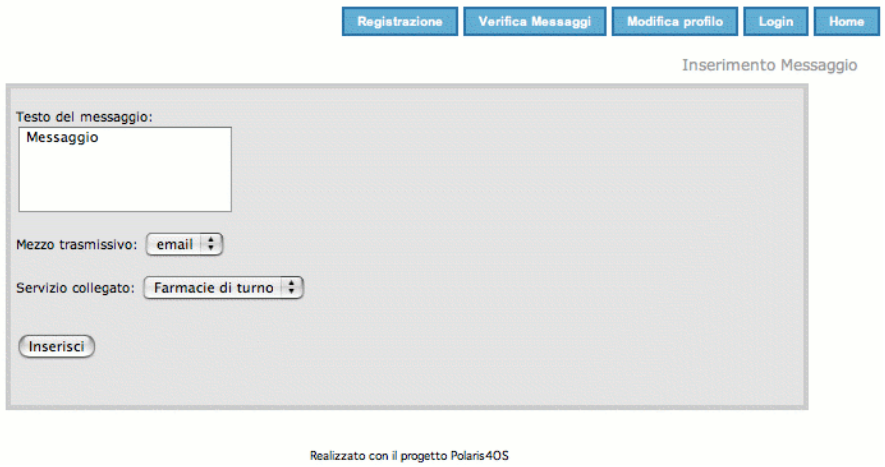

Figure 1: The web page to insert messages.

The tracking module is related to the European Union directive on the traceability of food products. In the case of a pigs farm the batch is identified by a code related to the week when the pigs born and it lets to identify their diet and medical treatments. Another customer request was to store the production costs. The developers studied the default database then modified the original one. Using the Compiere tools, the mask to insert data was created and integrated in the window management system of application.

2.5 The Companies Participation

Companies involved in Polaris4OS come from different IT market areas and they have different size. Most of them are focused on services for Public Administration, while others work in the Internet context, with web services and application, mobile solutions, digital television and professional training. Average team size is 3-4 developers for each company.

Before starting the project, each company presented itself to other participants, in order to better identify resources provided to the project and expected results. Technical team had experience in database engines, object oriented programming language and some web framework experience. Therefore, there was not an homogeneous expertise nor similar objective for those companies. Most important, there were no previous experience with open source projects.

CTOs have been introduced to F/OSS ecosystems, in order to provide them with all elements to decide what technology to adopt and what kind of OS project to work for. They have been preliminarily introduced to F/OSS licenses, their scope, their limits and their adoption across leading open source projects.

Companies have been encouraged to participate Polaris4OS because they would perceive the training phase as a valuable benefit for their team: they were able to gain new skill at no cost. However, they kept some doubts regarding actual developers

effort: it was expected to be two days a week and they considered it too short to achieve a relevant product but, on the other side, they were unable to assign more resources to the project.

3 The Curricula

An important and none evident aspect of the Polaris4OS project was the definition of the developers curricula. The teachers, tutors and coaches guided their actions in the designing and development phases in way to fit it.
The curricula definition started with the training phase and it was adapted to achieve four main goals:

- all developers had to work on the project;
- all developers had to use the selected technologies;
- collaborative approach to all project activities;
- build a developers shared knowledge, practices and learning community for all the following project phases.

We defined a list of fundamental competences for the curriculum and grouped them in four categories:

- information seminar;
- software engineering methodologies;
- use of tools and IDE (i.e. Java, Eclipse and Sourceforge);
- features and constraints of the two selected open source projects (local administration vs. tracking system) .

Defining curricula was based on what can be used to design courses and curricula concerning Agile development [10].

3.1 Defining the Curricula

The preliminary analysis identified the training goals selecting the areas of knowledge and skills the teachers, coaches and tutors needs to focus the lessons and to plan what designing and development aspect had to be treated.
The output of this activity has been the definition of the curricula based on the needed skills and started from the owned skills in way to fill the competence gap. Developers meeting evidenced they needed to keep confidence with development methodologies like advanced development practices (i.e. design patterns), database management, data abstraction and web applications. They had not experiences on software development in distributed environment so they did not know practices like test driven development (TDD), continuous integration, unified modeling language (UML) design and related tools. Furthermore the developers need to study a ERP and a CMS. In Table 1 are listed the identified modules and their correspondent categories.

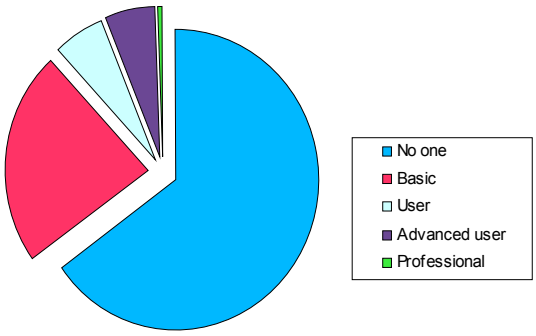

Figure 2: The graph exposes the initial developers skills.

3.2 Results of the BLM

The developers participated actively to activities on Moodle platform and its knowledge base. It is interesting to note that they assumed a proactive pose among the community. Many people had loaded appropriate material for other formers, they share informations, suggestion and doubts with on line coach and tutors. Problems was solved all together according to the collaborative learning. They appreciated the platform tools as much as web forums and chats.

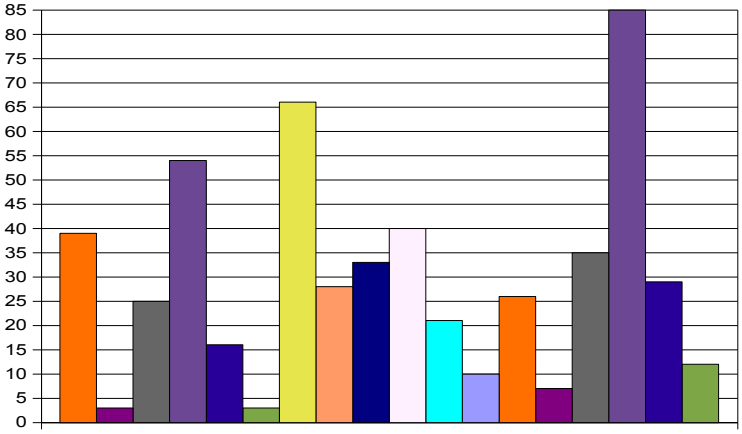

Figure 3: Number of visits to Moodle platform for each developers.

Module Name	Category
What is Open Source? - Open Source licenses and philosophy	Seminar
Collaborative Environment and LMS - Moodle and its tools	Seminar
Eclipse Base - Basic use of Eclipse	Tools
Java Advanced - Design Patterns	Tools
Eclipse Advanced - Developing application using Eclipse and CVS	Tools
UML and tools - Diagrams	Methodologies
TDD e Refactoring - TDD, Junit, HttpUnit, Frequent release, Short iteration, Refactoring	Methodologies
Web Application Development - Tomcat, Velocity, JSP,	Tools
DB and Data abstraction - Data persistence, DAO, Hibernate.	Tools
Distributed architectures and programming - Client/Server programming, Java network programming, XML standard	Tools
Infoglue CMS - System architecture and technologies analysis	Open source software
Compiere - System architecture and technologies analysis	Open source software

Table 1: The curricula modules and their correspondent category.

At the end of the third executive phase, a test was submitted to developers to collect their sensations on the whole project. The 82% of them provided a positive judgment: the 62% evaluated the learning process as “appropriate” and the 20% considered “appropriate at all”. Nobody found it “inappropriate”. On such results, we can assert the contents tough were appropriate to our target.

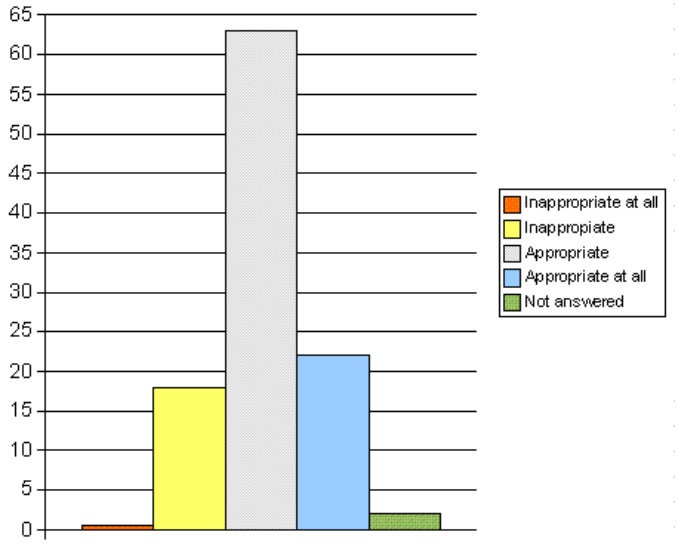

Figure 4: The results of the final test.

4 Conclusions

Same works have exposed the use F/OSS like a learning and model environment, for example [11], [12]e [13].
The results of Polaris4OS enforces such considerations. It has been successfully completed and it can be considered a very interesting case study to evaluate overall impact of F/OSS adoption in SMEs. Both company managers and their developers appreciated the F/OSS model and the concrete opportunities it gives. Working with real-world open source projects, they learned they can enhance their business, products and skills by means of F/OSS methodologies and practices. However, migration to F/OSS requires a focused effort to be understood, accepted and implemented.
The learners participated actively to curriculum definition negotiating topics with teachers in order to both satisfy project needs and reach their own expectations. It was a sort of negotiation about the pros and cons of each proposal and we want highlight that the last two curricula modules, focused on Compiere and Infoglue, were required by CTOs and developers.
Other works have treated F/OSS

Acknowledge

Polaris4OS was a project supported by Consorzio 21, funded by the European Union and performed by four executors: MATE, University of Cagliari, CRS4 and ApritiSoftware!. We want thank the developers, the coach Antonio Cuccu and Davide Onnis and Giuseppe Serra by Consorzio 21.

References

[1] McNutt L., Brennan M., , Learning Styles and elearning, what is the Connection?. FIE'05. Proceedings 35th Annual conference, USA, 2005

[2] Bunse C., Grutzner I., Ochs M., Peper C., Steinbach-Nordmann S., Applying a Blended Learning Strategy for Software Engineering Education. Proceedings of the 18th Conference on Software Engineering Education and Training (CSEE&T), Ottawa - Canada, 2005

[3] Beaton.C, Evolution of ethics blended learning. FIE '05. Proceedings 35th Annual Conference, USA, 2005

[4] Moodle. http://moodle.org

Shin-Wei Chou, Chien-Hung Liu,Learning effectivness in Web-Based Technology-Mediated Virtual Learning Environment2005

[6] Infoglue. http://www.infoglue.org

[7] Compiere. http://www.compiere.org/

[8] Rogovin A., Learning by doing: home and school activities for all children, Abingdon Press, 1998

[9] Eclipse. http://www.eclipse.org

[10] R. Camplani, A. Cau, G. Concas, K. Mannaro, M. Marchesi, M. Melis, S. Pinna, N. Serra, A. Setzu, I.Turnu, S. Uras, Using Extreme Programming in a Distributed Team . Workshop: Sharing Experiences on Agile Methodologies in Open Source Software, Genova - Italy, 2005

[11] Crowston K., Annabi H., Howison J., Masango C., Effective work practices for FLOSS development: A model and propositions. Proceedings of Thirty-Eighth Hawai'i International Conference on System Science (HICSS–38), Kona - USA, 2005

[12] Lehmann F.,FLOSS Developers as a Social Formation, http://www.firstmonday.org/issues/issue9_11/lehmann/.

[13] Rishab Aiyer Ghosh, Free/Libre/Open Source Software as a learning environment. Proceeding of International Symposium on Open Source Software, Abano Terme - Italy, 2005