MSWL Project Management: OLEF Project

Sergio Arroutbi Braojos

March 16, 2013

Contents

1 Introduction

WebTools SL is a company specialized on Web Developement. With a total of 20 employees, the company has developed different web tools designed and customized for different software development customers in Spain. Among the customers, they have important companies in different environments, such as:

- Hardware development companies.
- Software development companies.
- Internet Service Providers.

The main strength of the company belongs to the well qualified work force. For working in the company, having a good knowledge of Web development languages and tools is a MUST, although additional knowledge having to do with FLOSS projects associated to Web development are appreciated as well.

Moreover this, the company invest up to 10% of the earnings in technology training for the employees, in order to provide employees with updates of the latests technologies used, both in Web development, but also in other environments such as DDBB (DataBases) programming and administration, generic programming languages, scripting languages, software forges and VCS (version control system), graphic design, etc.

Among the weaknesses of the company, the lack of a strong marketing knowledge is an issue. Besides this, the commercial work force consists only of 2 of the 20 people working on the company, so selling strength needs backing to wide spread the company's business.

Among the different available web products they have developed, main incomes come, nowadays, from most to less importance, due to the next ones:

- Web Network Administration tools.
- A web software forge frontend for CVS and SVN.
- A web forum framework, based on CSS3, HTML5 and JQuery.

All of this software products are under propietary licenses, with copyright belonging to the company.

Regarding Web Network Administration tools, the product is well sold and has a strong market share position, increasing along the time, so no changes are considered in terms of licensing, development strategy or billing. Meanwhile, the company has lost definitely the market associated to their software forge web tools, due to the appearance and huge spread of GIT.

Somehow, this company wants to promote, improve and enhance their Web Forum framework, called **OLEF** (Open Libre Enhanced Forums) in order to explore this Market Niche. Feedback regarding this tool is quite good from customers perspective, above all in terms of usability, performance and look-appearance.

However, customers have communicated the company the lack of different functionalities they would like to own for integrating the product on their corporate network. Among the different functionalities pending, next ones can be found:

- Integration of an additional Question / Answer mode, similar to Stack Overflow
- Integration of a badging/vote system to highlight the best responses
- Integration of a much powerful report & statistic tool
- Integration of a live chat and voice over IP communication system between the forum members
- Security improvements
- Multiple theme selection

The mission is to achieve previous enhancements in one year time.

Unfortunately, the company has no budget to invest on developing all the new functionality, so releasing the software as FLOSS is the unique solution for achieving the goal.

The company has a total budget of 50000 to invest on the project, although has clarified that this budget should be invested on marketing, communication channels and in general for the project promotion activities. The company is not willing to contract additional developers, although will contribute a project leader and three developers to the OLEF project.

Two very important customers would be willing to integrate the FLOSS product and acquire a very important integration and support contract with the company.

So, to summarize, this is the challenge. Developing all the new functionality described above by starting a FLOSS project, with no impact on performance, while keeping the look and feel and ease of use of the framework.

2 Current competitors analysis

Web Forums are a very useful way of off-line communication. This kind of web interaction has reached to be the standard for handling question/answer issues related to a specific topic. FLOSS world is not an exception to this point. Rather, it can be ensured that FLOSS projects use and deploy web forums even in a more active way compared to other fields.

Apart from previous statement, it is important to highlight that, in the last years, new Web programming languages such as HTML5, together with new version 3 of CSS and libraries such as JQuery have improved drastically the web look and feel.

However, it seems that in Web Forums market, no further development of Web forums has appeared based on this technologies, or if existing, there is still a market niche to work and take profit on.

On this chapter, some of the main Web Forums frameworks will be analyzed, in order to consider the strenghts and weaknesses of any of them and take some conclussions on the different aspects where development of the project should concentrate.

Both FLOSS and propietary software Web forums would be analyzed, although the first will be the ones to focus on, as they are in fact a closer competitor due to the similiraties of strategy followed (in terms of community management, distribution, business model, etc.).

But, which are the most used and deployed Web forums existing on the Internet? Although no reliable market share research have been found related to this issue, some of the most important forums can be checked on this URL: `http://rankings.big-boards.com/`

This web indexes main Web forums around the world and ranks them considering the number of posts existing on the web. Although not a 100% trustable method, is a good approach to consider how spreaded is a particular forum. So, to summarize, `http://rankings.big-boards.com/`
contains information of the main Web forums all over the world, such as:

- Number of posts
- Number of users
- Statistics of the site
- Language used
- **Software used** (when it was possible to identify it)

A quick inspection on the data hosted on this page allows to take some conclussions. On the one hand, it seems that Web forum software market is basically splitted into two main Web forum frameworks:

- **vBulletin**. A propietary software which is, apparently, the market leader on Web forum market, with a total of **1384** entries on the web.
- **phpBB**. A FLOSS web forum software which a wide-spread use on the Internet, with a total of 240 entries on the rank.
- **Invision**. A propietary web forum software, with a total of 226 entries on the rank.

The rest of the forums have a much lower market share, and no further inspection will be performed. Previous data allow to take some conclussions regarding the market:

1. The market leader solution is **Propietary Software**.
2. The second market position, although being a FLOSS project, could be considered to have a very **much lower market share compared to the market leader**.
3. In the third position is, again, placed a propietary Software Web forum framework.

In next sub-sections, a deeper analysis of each of the two first software solutions would be performed, in order to identify the weakness and strenghts of each of the products, one of each type (FLOSS and proprietary software) and take some conclussions on the development strategies.

2.1 vBulletin

vBulletin seems to be the leader in the Web forum software. Developed by Internet Brands Inc., some of the main world-wide forums, such as `http://offtopic.com` are using this software. But, why is this software so popular?

Why does it have a so big market share compared to its competitors? What is the key factor for the success of this software?

No evaluation can be provided on this forum software, as it is a non-free (249$) propietary solution, but, having a look at the most important forums using this software, the software seems to be characterized by **strenghts** such as:

- Flexibility: A multiple bunch of plugins allow to improve the functionality and, above all, to adapt the look and feel to the administrators necessities.
- Ease of use: In terms of user experience. The interface for the end user is really simple, but, in turn, very complete.
- Ease of administration: In terms of administration permissions, banning, statistics, post management.
- **Look and feel**: The key factor. It is, surely, the best in this aspect.
- Robustness: In terms of bug, errors discovered. The up-time statistics are also very good in these aspect.

- Integration with Content Management Systems, such as Wordpress.

Meanwhile, the main **weaknesses** of this product are related, basically, to the fact of **not being FLOSS**:

- No inspection of the software can be performed.
- No modifications can be performed to adapt to the user requirements.
- Support depends on a single-vendor. No FLOSS community exist associated to the product.

Apart from that, having to pay for the license make it not available for certain communities with no budget for software acquirement.

2.2 phpBB

phpBB is the leading FLOSS software for Web Forums. Some of the most important forums world-wide, such as `http://gaiaonline.com`, are hosted on top of this forum software framework.

In terms of software, the product seems to be characterized by **strenghts**, some of them associated directly to being FLOSS, while others don't, such as:

- Community Support
- Developer Support
- End-user Ease of Use
- Robustness: In terms of bug, errors discovered. The up-time statistics are very good, and the community behind allows continuous improvement in this sense.

Meanwhile, among the **weakness**, it can be highlighted the next ones:

- SEO-friendliness
- Installation/Administration Ease of Use
- Flexibility Due to the difficulty of installing Plugins/Themes
- CMS integration

To summarize, it has to be highlighted that there is an important market niche in Web forum market. The key factor is trying to improve those factors that are weaknesses of phpBB FLOSS project, such as:

- Ease of installation and administration
- Ease of module expansion and plugins
- CMS integration

- Look and feel

Taking into account that the Look and Feel issues are well considered in our product, for being programed in HTML5/CSS3/Jquery and PHP, it is concluded that the main development strategy will be focused, on the one hand, **on administration/installation ease of use**, and, on the other hand, **on module and themes expansion** and integration in general, and with CMS (Wordpress preferably) in particular.

3 Licensing selection

Licensing model is a key factor that must be deeply analyzed in order to decide different factors which will be adopted up on creation of the OLEF open source project, and will continue to follow the project along the time.

3.1 Opensource vs. propietary software

From a company perspective, community involvement is a very important factor, and quick grow of the community is a desirable fact that should be maintained along the time.

Apart from that, changing OLEF to be an open source project and release under an open-source licensing schema is not a desirable strategy, but the only path that the company could follow in order to accomplish the goal.

Apart from previous statement, **no considerations of releasing under any propietary license have been taken**, mainly because of two reasons:

1. **Community and opensource commitment**. A high risk is observed by the company on considering any kind of propietary software licensing, even with a dual licensing model.
2. **No necessary risk**. From the company's perspective, no considerations on building a business model based on a propietary licensing has been considered. Even an opensource business model based on an open core and propietary licenses add-ons is not necessary, as main customers have communicated the company their desire to focus on having a service contract associated to integration, support and training services.

3.2 Licensing model analysis

Once the licensing schema has been decided to be based on a fully opensource licensing model, the main decission consider is which license will be selected for the project. In order to decide which opensource license to pick, some requirements of the project must be considered:

- **Small company**. This project is born in an SME, with very limited resources. For this reason, the license selected must be one of the most popular licenses in the opensource environment. No license proliferation can be started by the company, neither a not known license can be selected due to the doubts it can suppose for particular contributors to enter OLEF project community.
- **The fastest widespread of the project, the better**. Regarding community involvement, it is preferrable to select the most accepted license from the community perspective.

- **Everybody is welcome**. No particular target user/developer/business are considered to be part of the OLEF community. The company commitment to the opensource model makes that other companies start contributing a desirable fact along the time. In this manner, very different beings, from particular developers to big companies, will be welcomed to contribute and participate on the OLEF project.
- **Best knowledge here business model**. Taking into consideration the business model to be acquired, oriented to be based on support, training and integration of the solution, it is not a risk the creation of forks of the project to create a propietary product based on it. The major risk considered is the fork of the project to another opensource project. To avoid it, approval of the OLEF project as an opensource project from the community will be a mandatory desirable fact.

Having in mind previous requirements, **the best option seems to be a copyleft license schema** based on the GPL license, that will provide Community Involvemen, due to the fact that GPL is the most used license. However, copyleft model in general, and GPL license in particular, are not well considered by some of the potential community members, in particular two types of them:

- Some community members: Some community members of very important projects such as Apache, Mozilla, or BSD-like distributions, consider copy-left strategy as a non desirable one due to the obligations that this kind of model implies in terms of licensing compatibility and contamination.
- Some companies: Whose business model could be associated to split the project or part of it to a proprietary business model, or just to a non copy-left model.

For previous reasons, an alternative approach to license the code is based on a **dual licensing model**. This involves releasing OLEF project software under two licensing schemas:

- **A copyleft licence**. In these terms, a strng copyleft license, such as the GNU GPL, would be considered in order to commit with the community, favour its involvement and wide spread the project, the more, the better.
- **A permissive license**. In order to allow users or companies who are not intended to be bound by the copyleft license to contract the non-copyleft version. Generally, they would do this because they wish to produce a product based on the code but not confined by a copyleft licence.

Once the license model has been considered, it is very important to select which GPL license and permissive license would be considered.

Regarding GPL license, **GPLv3** would be selected, due to the fixes that this version represent compared to the previous version (GPLv2), in terms of patent closures, and compatibility issues.

Meanwhile, regarding the permissive license, the main requirement will be the selection of a well known permissive license with patent closures. For this reason, **Apache License v2.0** would be selected.

4 Technical Infrastructure

On this chapter, a detailed description of the technical infrastructure needed for the project will be exposed. Before inspecting the different alternatives available to deploy the project, there are some basic requirements that must be accomplished by the platform selected. Among the necessary tools that the , it can be found:

- **Software forge**. Based on GIT, due to the importance of this CVS, as well as the increasing popularity it is acquiring, even more on FLOSS projects.
- **Bug/New Functionality Tracking**. A flexible and ease of use based, integrated with the software forge. It is considered, by company previous experiences, a very important issue developers can check which changes correspond to a a certain bug or functionality.
- **Documentation**. It is a desirable aspect to have a wiki integrated on the software forge, which allows to link to documentation of the project from the source code.
- **Code revision**. Integrated if possible in the software forge platform.

Apart from the software forge, other communication channels will be desirable to contribute, from the project management point of view, to facilitate the communication between all the participants in the project. To do so, next resources will be available for the people involve to communicate each other:

- **Mail Lists**.
- **Real Time Communication tools**.
- **Forums**.

Regarding those **aspects that will not be managed by the company** because of its budget limitation and small size, will be:

- **Internationalisation**. English will be the default language. If translation contribution are provided by volunteers, will be welcome, but no further effort associated to translation is considered by the company.
- **SQA**. Specific SQA task force will not be assumed by the company. A basic test will be performed for each software release, but no regression test suites will be developed on the company. A continuos integration system will also be desirable for the company, but just to check and validate the code, not to perform deep testing tasks.

Last, but not least, there are some requirements that the company has introduced to the project managers:

- **No budget for technology platform available for the OLEF project**. All the hosted technology platform services must be cost-less. Company will only contribute with some servers that would be used for internal purposes (basically, continuous integration).

- **Ease of use**. Ease of use will be a MUST for the platform technology. Company want the fastest development, the better.
- **Look and Feel**. In order to continue with the strategy of high look and feel quality, very simple and not customisable platforms would be better not used.

On the next chapters, different technology platform would be analyzed, in order to take a decission of which one to use taking into consideration previous requirements.

4.1 Study of the main software forges

With requirements and considerations ennumerated , the most famous cost-less technology platforms will be analyzed, to check to which grade the requirements are accomplished:

1. **Launchpad**. Although Launchpad is one of the best cost-less FLOSS software forges on the Internet, it uses Bazaar as preferable VCS. Although using GIT is somehow possible on Bazaar, seing how committed to Bazaar is Launchpad, and taking into consideration **the requirement of using GIT as VCS, Launchpad would be preferred not to be used**, at least, before analyzing if there are other options more commited to GIT VCS.
2. **BerliOS**. BerliOS Developer is a costless service to FLOSS projects offering CVS/SVN/Mercurial/GIT VCS, as well as, mailing lists, bug tracking, message boards/forums, task management, etc. However, the **small spread of this platform**, that hosts less than 5000 projects and just a bit more of 50000 registered users. These numbers are considered a **risk in terms of community support related to the platform**, so this reason is enough to avoid this technology.
3. **GNU Savannah**. This software forge will not be selected due to similar reasons compared to the previous one. 58421 registered users and 3505 hosted projects make the company to consider **the small spread of this platform**, so this reason is enough to avoid this technology. Moreover this, other aspects such as the low ease of use as well as well as the not desirable look and feel make this platform even less recommendable compared to any of the previous ones.
4. **Bitbucket**. Bitbucket is a software forge belonging to Atlassian Software company. With a very nice look and feel, this software forge is based on GIT and Mercurial CVS, and offers several costless . However, the fact that no all the functionality is available costless make this forge preferably to be avoided.
5. **Google Code**. Google Code software forge has a complete source forge with different features such as code revision, bug tracker and wiki. The VCS is compatible with GIT, Mercurial and Subversion. However, look and feel is excessively simple. However, Google Code is contemplated as an option to be used as source forge, even more with a company as Google INC. involved.
6. **Source Forge/Allura**. Source Forge is the market leader in terms of projects hosted. With more than 400000 projects hosted, well-known FLOSS projects such as

VLC Media Player, Apache OpenOffice, FileZilla or TortoiseSVN. With GIT CVS, a nice look and feel and its market position, is another option for the OLEF project.

7. **Github**. Github is the other main software forge available on the Internet. With an increasing spread since it was initiated in 2008, the company has increased its popularity on an exponential way. With a plenty of nice functionalities, a commitment to GIT that is present even on the platform name, a very acceptable look and feel and a continue improvement philosophy, Github is the last selectable option to host the OLEF project.

At a first glance, any of the last three previous solutions should be valid in order to host our project. However, Google Code look and feel is a handicap for this forge to be selected, and decreases the real options to Github and Source Forge software forges.

In the following subsection a deeper comparison of both software forges would be performed, in order to have a more detailed information when taking the definitive decission on which platform should be selected.

4.2 SourceForge vs. Github

The previous chapters have helped on filtering among the different and more important platforms available to implement a source code forge platform, resulting on two candidates: SourceForge and Github.

Which is the more suitable platform for OLEF project? To carry out this decission, a first comparison of the different functionalities of each platform will be exposed.

Next table, shows the differences in terms of available features of each of the software forges under analysis, in particular, for the most important issues considered for the OLEF project:

Forge	**Code review**	**Bug tracking**	**Wiki**	**Web hosting**
GitHub	Yes	Yes	Yes	Yes
SourceForge	No	Yes	Yes	Yes

Bringing to foreground the different requirements exposed on this chapter, there is an issue which is not handled on SourceForge and can be managed in Github, which is **code revisions**. The lack of code revisions on SourceForge, together with the other requirements previously analysed, make **Github to be selected as the software forge for the OLEF project**.

4.3 Additional project tools

Selection of the software forge platform technology, in particular Github, make some of the requirements of the project to be accomplished. Among these requirements, next ones

should be highlighted:

1. **Software forge**. Distributed software VCS based on GIT.
2. **Issue tracking**. Integrated with the VCS.
3. **Wiki**. In order to collect most of the project documentation, the Github wiki will be used.
4. **Code Revision**. In a "pull-request" manner, Github allows a code revision mechanism, together with a system that enables code comments on each of the changes.
5. **Web Hosting**. Main web page will be available on Github to be the central point of access to the different sections associated to the project.

However, some other tools need to be introduced to complete the collection of applications for the project to be successfull.

Taking into account requirements of the project, basically, **no budget for platform technology**, the rest of the tools to be used by the company staff and project community will be exposed.

On the technical plane, a pair of **continuous integration servers**, for syntax check handling, **based on Jenkins project**. These servers will exist on the company on a 24x7 basis to perform a very light SQA. These servers are costless for the company, as they are part of the spare hardware availabe on the company.

Meanwhile, on the other hand, **tools focused on communication channels and social media must be implemented**, in order to ensure flowing communication for the project community. Among the communication tools considered, next ones will be created:

- **Mail Lists**. Some mail list will be created through "MailMan" site to . In particular, next three ones will be created:
  - **olef-users**. Where all the people who uses the OLEF Web Forum on a user way will be able to send doubts, critics and other issues.
  - **olef-administration**. Where all the people who uses the OLEF from an administration role will be able to send doubts, critics and other issues.
  - **olef-developers**. Where developers can send development issues, ask for advise on development tasks, etc.
  - **olef-management**. A private list where the project managers can discuss on OLEF project administration matters.
- **Real Time Communication**. Real Time Communication will be implemented by the creation of a IRC channel. In particular, the channel **#olef** will be created on the IRC node **irc-freenode.org**.

- **OLEF Blog Planet**. There will be a **main blog to keep track of the main news** related to the project, such as new software releases, important customer achievements, important milestones reached, etc. Meanwhile, by means of a **blog planet**, developers and project managers will be able to create their own blogs in case they want to comment on different stuff related to the project.
- **Forums**. Obviously, **implemented through the OLEF Web forum FLOSS project**. These forums will be both for "help me" issues, but above all for showing potential users the quality and revolutionary look and feel of the platform.

Meanwhile, regarding social media, next channels will be created:

- **Twitter**. By means of creation of a normal twitter account.
- **Facebook**. By means of creating a "Facebook page".
- **identi.ca**. By means of creation of a normal identi.ca account.
- **Linked in**. By means of creation of a project page associated to the already existing of the company.

5 Community Management

On this chapter, an analysis of the different decissions to be taken according to project management issues will be performed. To do so, this chapter will contain two main subsections to consider the different strategies to face by project management roles.

On the one hand, consideration of the **technical issues** will be proposed on next subchapters:

- **Organization politics**. In this subchapter, the strategy to face in terms of FLOSS project management approach would be considered. Basically, requirements of the project will be analyzed to check if the strategy must be more similar to the Apache Project one, based on Neutrality and Community driven, or, if on the opposite side, a more Single-Vendor strategy must be followed, similarly to the one followed by companies as MySQL or SugarCRM.
- **Development plan**. On this subchapter, the development strategy will be proposed. Which features of the project are most important? Which is the development focus? Considerations regarding development plan, roadmap and good practices for source code development will be considered.
- **Documentation**. This subchapter will collect which documentation should be created for the OLEF project, for each of the different roles involved
- **Emphasis on**. This subchapter handles with the strategy to follow in terms of project key aspects. Would be the project focused on Quality? Will it be focused on usability by end-user and administrators?. On this chapter answers to previous questions should be found.

On the other hand, next subchapters will bring to foreground the different aspects related to **community management**, marketing and communication channels:

- **Communication strategies and Marketing**. This chapter will focus on the marketing activities that will be performed, taking into account the available budget and the strategic decissions that were taken into consideration.
- **Volunteers: Community Management**. Which mechanisms will be implemented to attract community? This section will analyze different actions that can be taken to sort out this issue.
- **Netiquette rules**. On this chapter, the netiquette rules that will rule the community will be foregrounded, in order to keep a good environment and avoid unnecessary flames.

5.1 Project management

5.1.1 Organization politics

5.1.2 Emphasize on areas

5.1.3 Development plan

5.1.4 Documentation

5.1.5 Netiquette rules

5.2 Marketing and communication

5.2.1 Marketing

5.2.2 Communication strategies

5.2.3 Community building and new volunteers attraction

6 Conclussions

Releasing a project is not a trivial task. On this report, some of the different aspects to be considered when an SME company takes the decission of releasing an existing project as Open Source have been analyzed.

On the introductory chapter, an analysis of the company, the company's environment and the product to be released were brought forward. It is important to contextualize the different aspects where the project is located, in order to take decissions around the strategy that must be followed.

Subsequently, a deep analysis of the competitors was performed, in order to know the main competitors in terms of market share, and recognize the strengths and weekness of each of them in order to identify the strategies that the project will perform to try to get a good position in terms of market share, market acceptance, community involvement and other facts.

Afterwards, some considerations on the licensing scheme to be adopted by the project was foregrounded. In licensing selection chapter, some basic tips were introduced to show how selection of a good license or set of licenses is a key factor not only in terms of business model, but also in terms of community involvement and acceptance.

Technical Infrastructure was later studied in order to clarify which infrastructe fits better into the OLEF project requirements, not only considering the technical requirements, but also with an eye of the community necessities and how project management roles can supply different mechanisms to help on fixing them.

Last, but not least, a chapter having to do with the most important issues related to the project management were considered, not only from a technical perspective, but also from a marketing and community building perspective.

After the analysis of previous issues, the only goal of this report is to help on providing an introductory guide to some of the most important aspects to be considered when starting a opensource project, and the different decissions that could be taken to accomplish the project requirements and goals.