

# Project Management

What a project needs ?

Ricardo García Fernández

February 10, 2013

©2013 Ricardo García Fernández - ricardogarfe [at] gmail [dot] com.

This work is licensed under a Creative Commons 3.0 Unported License. To view a copy  
of this license visit:

<http://creativecommons.org/licenses/by/3.0/legalcode>.



# 1 What a project needs is

Firs of all,

- Mailing lists: Max
- Code: repository, VCS, Centralized, Distributed.
- Bugs: BTS - Launchpad bugs/issuses between projects.
- IMP: Integral Management Platforms. Redmine, trac. Github vs Gitlab.
- Licensing: copyright holders.
- Real-time comm: irc, mumbell, Jabber, identi.ca, twitter.
- Feedback: Forms, surveys.
- Forums (Avoid community fragmentation related to localization), blogs (Automatic Translations), planets (syndication) - Multilanguage.
- Project Documentation
  - Requirements
  - Design
  - Manuals
  - Faqs
  - HowTos
  - About
- Developers manual.
- Translation (Laura Arjona). English by Default.

Skilled people on the communication technologies

# 2 To unify efforts or not

- GNOME - KDE
- Beryl - Compiz: They share their goal, Desktop Effects.
  - Beryl focused in Configuration.
  - Compiz focused in power (KDE and Gnome development).

Se separaron en el camino y cayeron en una especie de ostracismo. Ms adelante Ubuntu used compiz as desktop effects base. Volvieron a unir esfuerzos despues de separarse.

- Several distributions (Debian, Suse, Fedora, Mandriva, Ubuntu, Linex, Guadalinux

### 3 Visibility

*Community building.* Attract new users, developers. Project visibility comes after improvement in development. *Near to visible projects:* be included in a huge project. You can lose autonomy or stilling, for being part of a bigger project. Prepare Scalability.

### 4 Structure

About & Contact. Basic info; *First steps:*

1. Download get-it.
2. FAQ.
3. Demo.
4. Doc.
5. Hot news.
6. Mailing lists, wiki, trac...
7. Related projects.
8. Users/Clients Portfolio.
9. Licensing
10. Link to VCS.

### 5 Installation process

Difficulties mean:

- Lost time answering doubts.
- Lost time fixing problems.
- Lost time updating the web site.

## 6 Cleaning code

- Your code is your image.
- Commits mailing list.

## 7 Files

Source code have several "burocratic" files:

- LICENSE
- README
- Changelog
- INSTALL
- Documentation (Users, developers, managers)

## 8 Demos

- Live demo environment it's the best.
- or data in a database, XML, CSV...
- Proof on concept

## 9 Tools

Keep record of everything...

## 10 Netiquette

Inline Reply in mails.

## 11 Ok, but where ?

What is it ? where to get it ? features requirements

## 12 Real case

*SourceForge*: when Github appears, they published Allura ALM. Python + Ruby.

*Forge Business models*: Break Interoperability to be the best. So the project won't change to another forge. **TODO**: Forges Interoperability Libresoft.

*Redhat,Novell*: service providers need a good translation and localization to avoid fragmentation.

*WebKit*: Automated Integrated System.

*Visibility*: GIMP, dump code and create Plugin Oriented Model.

## References

[1]