

# Participating in Free Software Projects

VALUG, 2012

Daniel Hofer

<http://dhofer.com>

Florian Preinstorfer

<http://nblock.org>

Wolfgang Silbermayr

<http://silwol.net>

10.08.2012



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Austria license (CC-BY-SA).

# Outline

Introduction

Bug Reports and feature requests

Contribute code

Resumé

# Why participate?

- Scratch your own itches
- Altruism
- Karma

# Types of project organization

- One-man-show
- Group of volunteers
- Organized and/or funded by organization
  - Foundation
  - Company (often with open core development)

# Detect level of project activity

- Mailing list
- Releases
- Source repository
- Forum activity
- Wiki history

For each of these, pay attention to age and frequency of activity

# Ways to communicate

- Mailing list
- Forum
- IRC
- Bug trackers

# Rules for communication

- Be polite
- Do not raise a claim
- Reply to the questions of the developers

# Types of contribution

- Bug reports
- Code patches
- Graphics (e.g. icons)
- Documentation
- Translations
- Answer to questions on forums and mailing lists



# Copyright Assignment

- Allows organizations to subsequently change license
- Controversally discussed
- Can be good (change from GPLv2 to GPLv3)
- Can be bad (company uses your code in closed-source software)



# Bug or feature?

- **bug**  
something that should work but does not work as intended by the developer.
- **feature**  
something which software does or would do if somebody coded it.

## Reasons for not being a bug

- The software should do something, that it currently does not
- A bug got already supported (show that you are affected)
- Inactive/deactivated functions
- Missing custom distribution patches that are present in other distributions
- Something is missing (.desktop file, ...) (distribution specific)
- if a package is out-of-date (distribution specific)



# Reasons for not being feature

- When it is a bug ...



## Search for existing reports

- Bugtracker of the distribution/project
- Mailinglist of the distribution/project
- \$searchengine
- Use the english version of the error message:

```
$ export LC_ALL="C"  
$ foobar --verbose
```



## Which version?

- Is the bug still present in the *latest* stable version?
- Is the bug still present in the latest *development* version?

## Distribution or upstream?

- If it is a distribution/integration problem: distribution
  - Missing dependencies
  - Problems caused by integration patches
- If it is a problem with the software itself: upstream
  - Crashes (segfault)
  - Broken/incomplete features
- It might be useful to report bugs on both bugtrackers

# What makes a good bug report? I

- Reproducibility
  - Steps to reproduce the bug
  - The expected behavior
  - The actual behavior



# What makes a good bug report? II

- Make the subject line a precise summary
- Include as much useful information as possible
  - Product version
  - Environment
    - Operating system and kernel version
    - Runtime environments (PHP version, Java VM, ...)
    - Your setup (config, web server, installed modules, ...)
  - Self-compiled or pre-packaged

# What makes a good bug report? III

- Attach relevant core dumps and log files
- If possible, use the included bug-reporting features

# Contribution process

- Find the project guidelines
  - Website
  - Contribute file in repository or source tarball
  - IRC / Mailinglist
- Coding style guidelines
- *One* feature / bugfix per patch
- Submit patch, communicate intention of the changes
- Track the proceedings of the patch
- If rejected, find out the reasons and improve accordingly

# Patches with git

- Create a patch using git:

```
$ git format-patch -1 #patch for the last commit with sign-off
```

- Apply a patch using git:

```
$ git apply 0001-name.patch
```

- Patch the traditional way (diff and patch): <http://stephenjungels.com/jungels.net/articles/diff-patch-ten-minutes.html>

## Code hosting platforms

- [github.com](https://github.com) (Git; Integrated bugtracker, wiki, release download)
- [gitorious.org](https://gitorious.org) (Git)
- [bitbucket.org](https://bitbucket.org) (Git, Mercurial; Integrated bugtracker, wiki, unlimited private repositories, custom domain)
- [sourceforge.net](https://sourceforge.net) (CVS, SVN, Bazaar, Git und Mercurial; Integrated bugtracker, wiki, release download, mailing lists)
- [code.google.com](https://code.google.com) (Git, Mercurial, and Subversion; Integrated bugtracker, wiki, release download)
- [launchpad.net](https://launchpad.net) (Bazaar; Integrated bugtracker, wiki, release download, mailing list)

# Contribution process on hosting platforms using DVCS

- Prerequisite: an account on the hosting platform
- Fork the project on the hosting platform
- Clone the forked repository locally
- Optionally start a new branch
- Develop the feature / bugfix
- Commit the changes to your local repository
- Push the changes to the forked repository
- Create a pull request for your feature
- Communicate with the developers (IRC, Mailing list)

# Resumé

- Be patient
- Stay motivated, even if it doesn't work out the first time
- Different projects, different rules
- Give back if you can



# References

- Arch Linux Reporting Bug Guidelines:  
[https://wiki.archlinux.org/index.php/Reporting\\_Bug\\_Guidelines](https://wiki.archlinux.org/index.php/Reporting_Bug_Guidelines)
- How to Report Bugs Effectively:  
<http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>
- The Ten Minute Guide to diff and patch:  
<http://stephenjungels.com/jungels.net/articles/diff-patch-ten-minutes.html>



# Participating in Free Software Projects

VALUG, 2012

Daniel Hofer

<http://dhofer.com>

Florian Preinstorfer

<http://nblock.org>

Wolfgang Silbermayr

<http://silwol.net>

10.08.2012



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Austria license (CC-BY-SA).