OSGeo

Your Open Source Compass

OSGeo Journal

Volume 6 - September 2010

VISITDENVER.com

VISIT DENVER, THE CONVENTION & VISITORS BUREAU

THE MILE HIGH CITY

DENVER

COLORADO

FOSS4G 2011 Denver, USA See You There!

2010 Official Visitors Guide

More information coming soon...

MAY 1 LIKE AN EGYPTIAN

with King Tut at the Denver Art Museum

Page 72

July 1, 2010 – January 2, 2011

JULY 10 A "WESTERN HEMIS-FAIR"

of art, culture and music at the Biennial of the Americas

Page 76

July 1 – July 31, 2010

From the Editor...

by Tyler Mitchell

Welcome to the first edition of the OSGeo Journal for 2010! As a good kick-off to the new year this volume takes a few different perspectives on software development and design. Naturally the various issues related to typical development projects applies quite well to our open source geospatial specific interests. The articles cover a range of topics from a review of various software to a discussion of user-centered design. Along the way you'll also get to read some more technically meaty articles and some perspective pieces.

Each volume of the Journal takes several months of concerted effort by many individuals. Landon Blake played a lead editorial role in getting this volume pulled together so you can read it - thank you Landon! It's always a pleasure to have more section editors, LaTeX masters and reviewers come to help. Thank you to all the volunteers.

With our new online management system, any potential article can be submitted at anytime by simply filling in a form at http://osgeo.org/ojs. As well, over the next couple of months keep one eye open for the OSGeo 2009 Annual Report. Get your articles in soon if you have not already. Enjoy the articles!

Tyler Mitchell
Editor in Chief, OSGeo Journal
http://osgeo.org
tmitchell@osgeo.org

Contents of this volume:

Editorial 1
From the Editor... 1

News 2
Brief News from the OSGeo Community 2

Event Reports 6
CASCADOSS International Symposium and International Information Workshop 7
Summer Training Courses on FOSS4G, 2007-2009 14

Topical Interest 18
Why Every Open Source Project Needs a Good Dictator 18

Programming Tutorials 20
GPGPU With GDAL 20

Case Studies 23
gvSIG is a viable robust alternative to commercially available GIS packages 23

Project Introductions 26
GRASS Image Processing Environment 27
SEXTANTE, a Free Platform for Geospatial Analysis 32

Peer-review Papers 40
Usability Trumps Features 40

Volunteer Recognition 53
53

Topical Interest

Why Every Open Source Project Needs a Good Dictator

How A Good Dictator Can Solve Problems With The Open Source Software Development Model

by Landon Blake

Introduction

This article examines the need for a 'good dictator' in open source software projects. It begins by describing the problems that can result in an open source project that lacks a good dictator, proceeds by explaining how a good dictator can help solve these problems, examines the qualities of a good dictator, and concludes with a description of the type of open source projects that would benefit the most from a good dictator.

Problems Resulting From a Lack of Clear Leadership

There are several problems that can plague an open source leadership that lacks clear leadership. These problems include:

Stagnation in development of the open source software that is developed by the project. Symptoms of this problem include infrequent releases and a lack of new features. Stagnation can result when there are no clear goals for the program or an overall push to take the program in a forward direction.

A 'muddied' code base that suffers from source code contributions with an inconsistent coding style. This coding style could be the formatting and structure of the actual source code, or the way the architecture of the program is designed and implemented.

Inconsistent branding of the open source project. This includes the design of logos, designation of text fonts and color schemes. Inconsistencies in these elements show up in both the documentation and the graphical user interface of the software produced by the project.

What is a Dictator?

Before we can describe how a dictator can solve the problems listed previously, it is important that we define the term dictator for the purposes of this article. A dictator is a person or organization that provides strong leadership of an open source software project. This leadership often comes in the form of an individual programmer, team of programmers, or organization that has specific goals for the open source project and a willingness to make decisions that may displease certain segments of the community.

The Best Type of Dictator

What makes a good dictator for an open source software development project? A good dictator has the

following characteristics.

A good dictator has a long-term commitment to the project. (For example: A company has a long term commitment to a software project because it distributes the software for a hardware platform that it produces. Another example: A non-profit organization that uses an open source program as a key part of its operations.)

A good dictator knows how to balance competing interests. This includes knowing what is worth fighting over, knowing how to best handle difficult people, and being able to tell contributors 'no' when it becomes necessary.

A good dictator has a large investment in the open source software project, but maintains an ability to see beyond their own immediate needs.

Solutions Provided By a Good Dictatorship

How does an open source software project benefit from a good dictator? How does a good dictator fix some of the problems that arise when an open source software project lacks good leadership?

A good dictator can set programming, marketing, and documentation priorities for a project. A good dictator can outline a development road map and set a release schedule.

A good dictator can set and enforce coding styles. This makes the structure and formatting of the actual source code more consistent. It also makes decisions about the implementation of the program architecture more consistent.

A good dictator can set and enforce standards for branding of the product. This can result in a clear image of the program's 'brand' and a more effective marketing effort for the open source project.

Landon Blake
Redefined Horizons
www.redefinedhorizons.com
sunburned.surveyor AT redefinedhorizons.net

This PDF article file is a sub-set from the larger OSGeo Journal. For a complete set of articles please the Journal web-site at:

http://osgeo.org/journal

Imprint

Editor in Chief:
Tyler Mitchell - tmitchell AT osgeo.org

Assistant Editor:
Landon Blake

Section Editors:
Daniel Ames
Scott Mitchell
Barry Rowlingson
Jorge Sanz
Micha Silver
Zachary Woolard

Acknowledgements
Various reviewers & writers

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the LaTeX 2_ε style source has been kindly provided by the GRASS and R News editorial boards.

This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this licence, visit:
http://creativecommons.org/licenses/by-nd/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

Journal online: http://www.osgeo.org/journal

OSGeo Homepage: http://www.osgeo.org

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8

ISSN 1994-1897

osgeo.org