

FSOSS: A Beginner's Perspective

Rueen Fiez

Date:

October 26, 2007

Student Name & ID:

Rueen Fiez-Moghaddam / 063777049

Table of Contents

FSOSS: A Beginner's Perspective.....	1
Table of Contents.....	2
1.0 Introduction.....	3
2.0 Summary.....	3
2.1 Code Reading and Review (Benjamin Smedberg).....	3
2.1.1 Background & Point of View.....	3
2.1.2 Thoughts About Code Reading & Reviewing in General.....	4
2.2 Product & User Experience Design (Mike Beltzner).....	4
2.2.1 Background & Point of View.....	4
2.2.2 Open Communities in Own Project.....	5
3.0 Views on Open Source.....	5
3.1 Open Source Views From Code Reading and Review.....	5
3.2 Open Source Views From Product & User Experience Design.....	6
3.3 Comparison.....	6
4.0 Personal Views on Open Source and The FSOSS Event.....	6
4.1 Post-FSOSS Impact On Own Open Source Project(s).....	6
4.2 Recommendations.....	7

1.0 Introduction

The main purpose of this report is to provide a concise summary of two presentations given at this year's Free Software and Open Source Symposium (FSOSS) as well as highlighting the speaker's view about open source and the surrounding communities in general. In addition, I'll be providing my own views about open source, its supporting communities, and what I was able to take away from the presentations given at this year's FSOSS. This being my first FSOSS event, I was naturally overwhelmed and somewhat intimidated by all the “big-shots” in the world of open source. I went into this event with one goal in mind - to gain a better understanding of free software and open source and be able to apply what I learned when working on my own open source related projects.

2.0 Summary

I chose to attend the following two presentations because I felt – based on their brief description in the agenda page – that they would provide me with knowledge that I could put to use throughout the development of my own open source project. My own project in particular entails much code reading, code reuse, and is supported by a relatively large Firefox localization community. After much consideration, I knew that the following two presentations, as well as their respective speakers, would best complement my existing knowledge about code reading, code reuse, and developing for communities – while adding an open source “spin” on them.

2.1 *Code Reading and Review (Benjamin Smedberg)*

Benjamin Smedberg's presentation outlined the reasons for reading code, difficulties encountered when reading code, code reading in a broader sense, and the different types of code review. He mentioned the various reasons why people read code which included – apart from it being an entertaining activity for “geeks” as he put it – fixing a bug, adding features, writing documentation, or just learning how a system works. Benjamin got down to the point and flat out said that code reading *is* hard especially when it comes to more complex systems. However, he emphasized that code in an open source context is not an individual activity and is in fact a social one. Code reading requires the participation of the code's author as well as the experts who contribute, maintain, or use the code extensively. The only exception to this rule would be if the system has no documentation, is relatively new, and the author of the code cannot be reached. The presentation also dived deeper into how to fix bugs by looking for common bad patterns such as “off-by-one” or “unsigned/signed mismatches”. I appreciated the advice he gave about mixing software frameworks together. Benjamin cautioned us that mixing frameworks, especially in JavaScript in particular, can lead to pitfalls in design and quality. In addition, Benjamin stressed that you do not have to understand the entire code, but only understand the part of the code that concerns to your end objective(s).

2.1.1 Background & Point of View

Platform Guru is the official title given to Benjamin Smedberg which came as a surprise to me since I've never heard of such a thing and it seems quite unconventional. Although I shouldn't be surprised, I've been told that Mozilla is a unique corporation in

many ways – which adds to its appeal. Benjamin's views on open source and its communities shouldn't be hard to guess, he emphasizes that open source as it relates to code reading and reviewing should be a collaborative endeavor. Not simply a lone coder sitting in a dark room straining his eyes over code they don't even understand – as Benjamin put it.

2.1.2 Thoughts About Code Reading & Reviewing in General

Based on what I have seen about open source so far, I believe that code reading is one of the most important skills a developer should be proficient at when working with the open source community. A developer who is able to read code and adapt it for their own purposes regardless of its complexity is able to employ a more versatile array of tools for each iteration. Code review has many benefits and applications, but personally, I perceive it as a skill necessary to deliver quality software to the community.

2.2 *Product & User Experience Design (Mike Beltzner)*

Mike Beltzner's presentation was focused on products and the user's experience with those products in an open community context. He essentially summed it up with his three “maxims”. The first being **listening to your community** which is basically focusing on what the community wants or needs. His second maxim was **leading your community** which is essentially to provide organization and structure to direct contributions from the community. Utilizing small focused teams is another approach to leading your community. Mike's final maxim is to **allow the community to play and experiment** with the product. Community feedback is extremely useful to ensure a higher quality of software. He also pointed out how Microsoft was making an effort to gather user's experience with their products by highlighting the “Happy face” and “Frown face” icons that appear in the task bar. Mike's presentation was filled with useful information about relatively new software such as CoScripter and TheCoop. His presentation successfully influenced me enough to give these products a run and see what the hype is about. I'm happy to say that I found the experience very enlightening.

2.2.1 Background & Point of View

Mike Beltzner is responsible for the experience anyone gets when they touch anything related to Firefox, such as the website, the add-on site or anything else related to Mozilla. Before I get to Mike's views on open source, I feel compelled to mention that Mike was probably one of the most colorful characters I had the fortune of seeing during the FSOSS event. Mike's view on open source is that if you are working on an open source project and you take the time to listen to your community, lead your community, and allow the community to experiment with your product, that your final product will be of high value and quality because it has been embraced by the open source community. Mike also emphasized how the open source community is chaotic and a priority for any open source project should be to bring order to that chaos and to take advantage of that community when it is acting more like a “chorus”. Mike sees the

advantages of gathering user data and how building a community around a product is the most effective way of sustaining the product over the long-term.

2.2.2 Open Communities in Own Project

Since I my own project involves working with an existing localization community I knew that Mike's presentation would give me more insight into how to work with them. The community in my own project is relatively large and is being led by a few Mozilla employees. They have been leading us with structure and organization by providing us with scripts, tools, contacts and possible approaches to reaching our main goals and objectives. Conversely, we have also been following Mike's first maxim, listen to your community, by adding and removing features according to what we believe the community would want. Throughout the development of our project, we tried to keep our tool simple and not overly complex to use. We knew that if we made a complex tool, the community would reject it and the project would not progress any further in the future. With the help of Mozilla's Axel Hecht and Michal Berman (localization) and support from our professors Dave Humphrey and Chris Tyler - who have helped us considerably by providing us with considerable guidance and contacts - we have been able to get our project off the ground and have built the foundation for a tool that the community can put to good use in the near future.

3.0 Views on Open Source

Both speakers primarily talked about open communities and how leading them can result in greater benefits for open source projects or existing open source products. Although both talks were about different topics, they had similarities when it came to how to manage and maintain an open community.

3.1 *Open Source Views From Code Reading and Review*

Developers must take into account that when they are writing their code, they should be writing it in a way that will make it simple for another developer to come in and read it and understand it in some ways. They can do this through documentation or other means to communicate the code and design decisions that were made throughout development. By making code easy to read, the authors can take advantage of the vast community that accompanies most open source projects by helping potential developers understand the code so they can attempt to contribute to it in some way. Also, it's important for open communities to be able to read the code because most projects will have bugs. Often, the community is the one that will find these bugs and rectify them through patches. Developers must make an effort to make their code easy to read so communities can review the code without stopping at every class or method and asking "why do we need this?". Open source communities also have a tendency to reuse existing code – something that I'm beginning to understand a great deal about. When you write code, there is always a possibility that your code may be used and adapted to someone else's project. Proper documentation can make this adaptation easier for the developer and increase the quality of the product.

3.2 Open Source Views From Product & User Experience Design

Mike Beltzner views the open source community like an unorganized and chaotic group of developers and enthusiasts who all want the product to add a new feature or remedy some bug. He sees the leadership's goal, Mozilla in this case, as turning this unorganized chaos into an organized chorus. He believes that as a result of turning chaos into a chorus, the product will benefit from it in the end. Mike is a strong believer in getting feedback from the community and allowing the community to express their ideas about a particular product. He believes that open source projects and existing open source products should have a streamlined way of getting community feedback, reporting bugs, and learning about the product in general. A good way of leading your community is to identify those who are making “smart” comments about a product then give them positive feedback so that they are more likely to contribute to the product in the future.

3.3 Comparison

Although Mike Beltzner and Benjamin Smedberg's presentations differed in terms of focus, they did however have one thing in common – how the open source community can impact a product. Benjamin's presentation focused more on how communities can fix bugs or add features to a product whereas Mike's presentation focused more on how to organize and maintain communities and harness their potential power to elevate products to a higher level of quality. Both speakers also centered their presentation around how communities should be led. Benjamin spoke about how code that is easily read can be reviewed more efficiently for potential contributors whereas Mike discussed how structure and organization can help lead communities. The only differences between the presentations was that Benjamin went more in depth into why code review by the community is beneficial and Mike talked more about creating a streamlined method of gathering user experience data and use that data to increase a product's quality.

4.0 Personal Views on Open Source and The FSOSS Event

This being my first FSOSS, I did not have a clue the impact the speakers were going to have on me or even how much of the material presented I would actually put to practical use later on. Afterwards, I'd realized just how much FSOSS had impacted the way I perceive open source projects. In particular, Mike Beltzner's presentation was very informative and illustrated many useful systems I had no idea about such as CoScripter, TheCoop. He also reminded me of some others I'd forgotten about such as Miro, digg, Flickr, and many more.

4.1 Post-FSOSS Impact On Own Open Source Project(s)

After reflecting about what I learned during the FSOSS event, I was able to approach my own open source project from a whole new perspective. Looking through the myriad of localization tools – which is essentially a collection of Python scripts, XML files, DTD files, and Properties files – I was no longer intimidated when reading through its many classes and methods. As a result of Benjamin Smedberg's presentation on code reading and review, I knew that to understand everything about the code is not necessary, and that I only need to understand

specific parts of someone else's code and use them accordingly. Also, as a result of Mike Beltzner's presentation about open communities, I always kept in mind how the Firefox localization community would react to the design and programming decisions I make throughout the development of the project. Mike's emphasis on open communities made me realize how much I can help potential project contributors by highly documenting every part of my project. Everything from documentation in code, references to locations where code was reused, documentation related to product design, community contacts, and step-by-step processes. Not that I hadn't already documented, but Mike's presentation gave me incentives to document the project as much as I could for the sake of the community. I knew by keeping the community in mind during the course of development, potential contributors would avoid common pitfalls when trying to read or reuse the code.

4.2 Recommendations

My first FSOSS experience was a positive one, filled with a lot of colorful and knowledgeable speakers. I was able to take away a lot of information from the presentations and successfully apply what I've learned to my own open source projects. I would definitely recommend going to next year's Free Software and Open Source Symposium to anyone – not just software developers – who even has a passing interest in open source applications, or is affected by the open source community in some way. It is a great way to meet the leaders in the community and learn new ways of approaching projects.