Open Source Developer Layer Assessment: Open Onion

Aminat Abiola Showole, University of Abuja, Abuja, Nigeria

ABSTRACT

Open Source developers play fundamental determinant role in the life of any open source project. This paper investigates developer motivation in contributing tirelessly to an open source project. Open source Onions were investigated and the developer layer modeled and validated based on ten case studies from SourceForge. Validation was based on Delphi’s four rounds of successive stages. Results show that 62% of the developers around an open source development project community are skilled programmers, and that Open source developers are largely motivated by web based development platforms with universal programming language such as Java and that Developers are mostly attracted to the GPL licensed software development project with high project publicity as could be tracked from the hit rate on the project website. Finally, the few Core developers (Project Administrators) of about 19% actually controls and oversees the affairs carried out by about 81% of many developers showing the prominence of Pareto80/20 Principle in Open Source Project development.

Keywords: Open Onion, Open Source Developer, Open Source Development, Open Source Software, Open Source Success

1. INTRODUCTION

Open Source development could be analyzed based on a number of key factors; ranging from developer roles to investigating the community around a project, the target audience and the programming platform support for an open source project. Interestingly, open source development is usually described with onion structure. Paradigm examples are Crowston’s Onion as discussed in Crowston and Howison (2003); Kumiyo’s Onion as found in Kumiyo, Yasuhiro et al. (2002)); Antikainen’s Linux Onion (Antikainen, Aaltonen et al. 2007) and Herraiz’s Onion (Herraiz, Gregorio et al. 2006). The Open Onion research has evolved a robust open onion model of open source development by merging the onion layers within the four earlier mentioned open source onions. With this approach, a unique open onion model has evolved from the consolidated un-validated earlier onion models of open source. The novelty of this research lies in the validation of the layers within the open onion model. This paper only provides a detailed assessment.
of the developer layer of the open onion. Table 1 shows the evolution of the open onion model and developer layer is prominent within the onion model of open source.

1.1. Structured Literature Review of Open Source Onions

The open onion model of open source was first conceptualized in 2008 where the author developed a six-layered onion model of open source as described in Showole, Shamsul et al. (2008). Progress on further research resulted in the development of only five-layered open onion model in which the observer and non-interest group layer (6th layer, in the earlier conceptual model) was now being captured at the external layer (5th layer, in the Open Onion Model). This has resulted into a 5-layered open onion model of open source.

This research presents, analyses, assessed and validates only the developer layer. In future research, each of the other remaining layers would be developed uniquely for better presentation, clarity and understanding of the Open Onion Model.

2. A REVIEW OF THE OPEN ONION MODEL

2.1. Description of the Open Onion Layers

The detailed layered descriptive presentation of the open onion model of Figure 1 is described in this section. There are five layers and developer layer is the second layer after project initiation layer is achieved. Table 2 presents the descriptions for each of the layers.

3. DEVELOPER LAYER PRESENTATION AND ASSESSMENT

The developer layer is the second layer of the open onion model as shown on Table 2. This layer is comprised of three major components representing the usual activities and interactions of open source developers. These activities are distinctly captured in Figure 2.

In the developer layer, methodologies, approaches and techniques being adopted by the project are clearly defined. The methodologies map the activities to the software engineering tools. The active players in the developer layer are the people (developers) themselves. Therefore, the presentation of this layer comprises of the people, communication channels, software

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Internal</td>
<td>Core Developer</td>
<td>Project Leader/Core members</td>
<td>Linus Tovald/Lieutenant</td>
<td>Core Developer</td>
<td>Project Initiation Layer</td>
</tr>
<tr>
<td>Developer</td>
<td>Co-developer</td>
<td>Active/Peripheral Developers</td>
<td>Coders/ Jenitars</td>
<td>Developer</td>
<td>Developer Layer</td>
</tr>
<tr>
<td>Maintainers</td>
<td>Release Coordinator</td>
<td>Active Developers</td>
<td>Maintainers</td>
<td>Bug fixers</td>
<td>Maintainers Layer</td>
</tr>
<tr>
<td>User</td>
<td>Active Users</td>
<td>Readers</td>
<td>Testers</td>
<td>Mailing list</td>
<td>User Layer</td>
</tr>
<tr>
<td>Most External</td>
<td>Passive Users</td>
<td>Passive Users</td>
<td>Readers</td>
<td>Plain Users</td>
<td>External Layer</td>
</tr>
</tbody>
</table>
Community Customers
[www.igi-global.com/chapter/community-customers/21212?camid=4v1a](www.igi-global.com/chapter/community-customers/21212?camid=4v1a)

Digital Recording Platforms and Integrated Performance Assessments in Second/Foreign Language Learning