



HAL
open science

BULB: Onion-Based Measuring of OSS Communities

Terhi Kilamo, Timo Aaltonen, Teemu J. Heinimäki

► **To cite this version:**

Terhi Kilamo, Timo Aaltonen, Teemu J. Heinimäki. BULB: Onion-Based Measuring of OSS Communities. 6th International IFIP WG 2.13 Conference on Open Source Systems,(OSS), May 2010, Notre Dame, United States. pp.342-347, 10.1007/978-3-642-13244-5_29 . hal-01056043

HAL Id: hal-01056043

<https://inria.hal.science/hal-01056043>

Submitted on 14 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

BULB: Onion-Based Measuring of OSS Communities

Terhi Kilamo¹, Timo Aaltonen¹, and Teemu J. Heinimäki¹

Tampere University of Technology
firstname.lastname@tut.fi

Abstract. Up to date information on the associated developer community plays a key role when a company working with open source software makes business decisions. Although methods for getting such information have been developed, decisions are often based on scarce information. In this paper a measuring model for open source communities, BULB, is introduced. BULB provides a way of collecting relevant information and relates it to the well-known onion model of open source communities.

1 Introduction

A company working with open source software (OSS) is often dependent on the developing community. Especially, when a product or service is sold the connection is obvious. In order to make business decisions, the need for up to date information about the community is clear. For example, the size of the community should be known, the activity of developers is interesting, and how easy penetrating into the community is should be found out.

Currently getting this kind of information is hard. Precise models for such have been developed. For example, *social network analysis (SNA)* [3] has been suggested to get a strict view to the community. SNA analyses a mathematical graph, where nodes are the members of the community and arcs model relationships between them. Different kinds of surveys can be given as another example of community information digging.

The industry does not seem to use such advanced methods today. On the contrary, to our knowledge the business decisions are often based on simple models, and, it is not unusual that the only two sources of information are the number of messages in discussion forums and the number of downloads.

In order to be adopted in the industry, metrics need to be instantly meaningful. We propose is a measuring model *BULB*, which relates the measurements to the well-known onion model [7] of OSS communities. The onion model is commonly accepted and it is easy to grasp, so BULB conforms to the prerequisites. The measurements are based on robots digging continuously information from various sources, like the discussion forum, the version control and the bug repository, i.e. the framework conforms to the rest of the conditions.

The rest of the paper is structured as follows. In Section 2 ways to measure open source communities are discussed. The BULB model for measuring open source communities is introduced Section 3 and applying it to an industrial community is given Section 4. The paper is concluded in section 5

2 Measuring Open Source Communities

The community behind an open source project is a key component that effects the success of the project. Information on the nature of the community is needed in order to make informed decisions on adopting open source software and to aid running a successful business based on open source components. Several different approaches have been suggested to provide support in the decision making ranging from easy to get to more extensive analysis.

Social Network Analysis: Any open source project can be seen as a social network of developers. The developers are linked to each other through different kinds of relationships that are created and maintained in OSS projects mainly by computer-enabled channels. The OSS community is thus seen as a graph with the developers as the nodes and the social relationships between developers as the edges. Social network analysis (SNA) [3, 10] can be used to study the community and its structure.

Business Readiness Rating: Business Readiness Rating (BRR) [2] proposes a method for assessing open source software. The goal is to get a rating on the open source software through four steps (1. quick assessment, 2. target usage assessment, 3. data collection and processing and 4. data translation) As BRR itself admits that phase three is the most time consuming and yields best results for mature projects, its value is somewhat limited to eliminating bad candidates. It also seems apparent that BRR is no longer being developed further at the moment.

Simple metrics: One way to evaluate the open source project is to measure some publicly available data that are easy to access and measure. Very simple metrics, such as the amount of downloads or the daily amount of discussion on email lists or the project discussion forum are used. Naturally the level of activity in the community shows if the community is still alive, but says very little on the product or the sustainability of the community on the whole.

Software use: Software use is naturally a popular metric albeit one that is difficult to measure reliably in the case of open source software. Numbers of downloads alone do not reliably tell about adoption [1]. Moreover from the business decision point of view, the community as a whole is interesting, not just the usage.

Surveys: Surveying community members is a suitable method for gaining information from different interest groups within the community [9, 4, 11]. However, surveys don't necessarily reach people, whose input would be most valuable. In addition, surveys cannot be used as a mean of continuous analysis as the likelihood of people answering them decreases over time.

Appearance: A common method of comparing projects and making decisions on the project to use is not based on any kind of measuring as there often is no time to undergo a vast analysis. The appearance of and the feeling one gets from the community are the driving factors instead of a more formal approach. Results beyond a blatant guess are needed, and therefore the need to evaluate the community further is apparent.

Onion-Based Measuring: Open source communities can be modeled with an onion model introduced in [7]. In the model each member of the community has a distinct role. The community is seen as an onion-like structure, where the most influential community members occupy the core layers, while the outer layers hold the less influential ones.

3 Constructing an Onion-Based model: BULB

In this section a measuring model for onion-based measuring of open source communities, BULB, is given. The structure of the community and how the community members fall on layers in the onion is valuable information about the community and its current state in making business decisions. BULB has been developed for this very purpose. The theoretical base of the model has already been introduced in [6]. In it, the traditional onion model for open source communities is substituted with two onions, one for the size of the community and another for the amount of activity on the onion layers. The traditional size

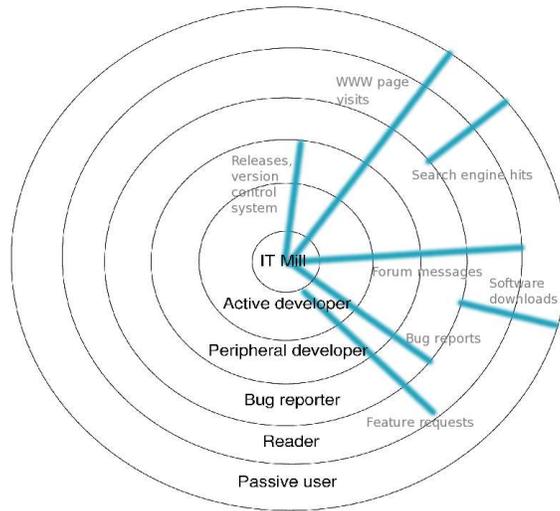


Fig. 1. The data sources on the onion

onion is produced by simply assessing the number of people on each layer of the onion. Data from several data sources is combined to get a picture of the

structure of the community according to the onion model. Data used at this point range from the version control system information to the number of web hits for relevant sites. Some of the data sources have more effect on the onion built out of the data than others. This is taken into account when constructing the onion model for the community. The data sources used and the onion layers affected by each source are depicted in Figure 1.

The onion is seen as a vector, where each element contains the relevant information about the layer, for example in the size onion the number of people on the corresponding onion layer. Each metric used is measured daily and a vector representing its distribution on the onion layers is created by multiplying it with a coefficient vector that indicates how influential the metric is on each layer. The coefficient values of the layers add up to 1.0. If we denote the set of metrics used with M the distribution vector \mathbf{d}_i of each metric is calculated:

$$\forall m_i \in M : \mathbf{d}_i = m_i \mathbf{v}_i \quad (1)$$

where \mathbf{v}_i is the coefficient vector of m_i . The example vectors used in the case study for distributing the numbers of bug reports and feature requests over the onion are shown in Figure 2. The different data sources can have significant

| | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-------------------|-----|------|------|------|------|-----|
| Bug reports: | 0.2 | 0.3 | 0.3 | 0.2 | 0.0 | 0.0 | Feature requests: | 0.0 | 0.25 | 0.25 | 0.25 | 0.25 | 0.0 |
|--------------|-----|-----|-----|-----|-----|-----|-------------------|-----|------|------|------|------|-----|

Fig. 2. Example coefficient vectors

differences in their relative values as one can be very large while the other occurs more rarely and is thus smaller. To compensate this the distribution vector of each metric is multiplied with a balancing coefficient b_i . We get a partial onion vector:

$$\mathbf{p}_i = b_i \mathbf{d}_i \quad (2)$$

The significance of the metric on the onion can also be scaled through this coefficient. In the example measurements, the balancing coefficients used were 7.0 for bug reports and 9.0 for feature requests.

After the balancing the complete onion is created by simply adding the values on each onion layer in the partial onion vectors together in order to create the final onion, i.e.

$$\mathbf{o} = \sum \mathbf{p}_i \quad (3)$$

The traditional onion alone is not able to accurately depict how active the community members on the different layers of the onion are but is simply focused on the size and structure of the community. The activity may vary over time although the size of the community has not changed. Thus the variation in activity on the different layers should be taken into account in addition to the development of the size of the community. As some of the metrics used may give information about the current level of activity on a given layer BULB suggests a second onion similar to the size onion to be used for depicting layer

activity. The activity onion is built like the size onion only based on the metrics that measure activity. The distribution vectors and the balancing coefficients are naturally adjusted suitably. In the example case, the balancing coefficients change to 100.0 for bug reports and 140.0 for feature requests as they are clear indications on activity.

4 Experimenting BULB with the Vaadin community

The BULB model is in fact a generic method of depicting the evolution of an open source community. The data values in the onion vectors can be changed to a new community characteristic and the model is still applicable. To study the applicability of the framework, we have experimented it with the developer community of Vaadin [8]. The measurements were carried out from May 1 2009 to Nov 30 2009. The onion model was instantiated to the Vaadin community as shown in Figure 1.

Vaadin is a server-side AJAX web application development framework developed by Oy IT Mill Ltd [5]. The framework is used for developing rich Internet applications with the Java programming language. Vaadin framework was released as open source in December 2007. The business model of the company is based on consulting services and the development of Vaadin. As Vaadin is open source, IT Mill needs up to date information about the Vaadin community. So far the main source of information has been the number of downloads and the number of messages post to the discussion forum. Figure 3 illustrates the

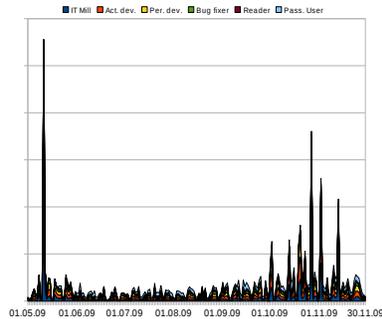


Fig. 3. The activity onion of the Vaadin community

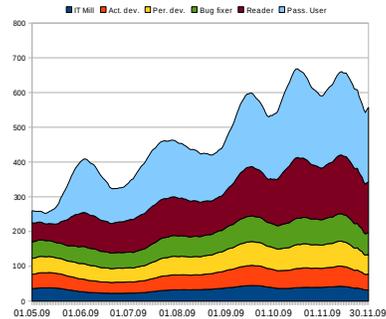


Fig. 4. Size onion of the Vaadin community.

measured activity in the Vaadin community over the measurement window. It visualizes the effect of events that have impact in the community. The size data however needs to be filtered to lessen the weekly variation in the raw measures. The size onion of the Vaadin community after filtering the data with a Gaussian filter is shown in Figure 4. The window size of the filter was 31, $\mu = 0$, $\sigma = 31/4 = 7.75$.

5 Conclusions

We have developed a new onion-based model, BULB, for measuring open source communities. We applied the model in an industrial case to measure the layer

sizes in the Vaadin community onion and the activity on the layers. The model was developed in cooperation with industry to make an easy-to-use and fast way for digging valuable information on an open source community out of the available data.

The board and other stakeholders of an open-source company or of companies thinking of adopting an open source product often base their decision to a limited amount of information. With BULB these decisions can be based on more fresh and divergent information than before. We have shown that the described model works and produces sufficiently precise information fast enough to be useful and support decision making.

References

1. Wiggins A., Howison J., and Crowston K. "Heartbeat: Measuring Active User Base and Potential User Interest in FLOSS Projects. In *Open Source Ecosystems: Diverse Communities Interacting*, volume 299/2009 of *IFIP Advances in Information and Communication Technology*, pages 94–104. Springer, 2009.
2. BRR. <http://www.openbrr.org/>. Last visited December 2009.
3. Del Rosso C. Comprehend and analyze knowledge networks to improve software evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, 21(3):189–215, 2009.
4. Capra E., Fancalanci C., Merlo F., and Rossi Lamastra C. A Survey on Firms' Participation in Open Source Community Projects. In *Open Source Ecosystems: Diverse Communities Interacting*, volume 299/2009 of *IFIP Advances in Information and Communication Technology*, pages 225–236. Springer, 2009.
5. Oy IT Mill Ltd. web page: <http://http://www.itmill.com/>. Last visited December 2009.
6. Heinimäki T. J. and Aaltonen T. An onion is not enough: Living in the multi-onion world. In *Proceedings of the Open Source Workshop - OSW 2009 In conjunction with the 4th IEEE Systems and Software Week (SASW 2009)*, Skövde, October 2009.
7. Nakakoji K., Yamamoto Y., Nishinaka Y., Kishida K., and Ye Y. Evolution Pattern of Open-Source Software Systems and Communities. In *IWPSE '02: Proceedings of the International Workshop on Principles of Software Evolution (2002)*, pages 76–85. ACM Press, 2002.
8. Grönroos M. *Book of Vaadin: Vaadin 6*. Oy IT Mill Ltd, 2009.
9. Ghosh RA, Glott R., Krieger B., and Robles G. Free/Libre and Open Source Software: Survey and Study. International Institute of Infonomics, 2002.
10. Wasserman S. and Faust K. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
11. Mikkonen T., Vainio N., and Vadén T. Survey on four oss communities: description, analysis and typology, 2006. Empirical insights on open source software business.