Evaluating the Health of Open Source Components

Ivan Zaytsev[a,*], Slinger Jansen[a]

[a]*Department of Information and Computer Sciences, Utrecht University, Utrecht, The Netherlands*

Abstract

Context: Implementing open source components into commercial applications has many advantages for software developers. However, an unforeseen decline in health of the supplying community can lead to a number of complications or large expenses, caused by transition costs to an alternative software component. Successful product managers must be able to assess the health of the open source communities their applications depend on.
Objective: In this paper we present a modular method for software product managers that allows them to assess the health and vitality of open source communities.
Method: The research is founded on a systematic literature review on the topic of open source and software ecosystem health, as well as a case study at a software firm with extensive open source experience.
Results: The main research result is an Open Source Component Health Analysis Method that can be applied and fully customised by software product managers. The method is based on a list of open source vitality indicators, as well as an open source interaction model, including the role of commercial patronage in contemporary open source communities.
Conclusion: Recent appearances of commercial patronage appear to dilute the classical distinction between voluntary private contributions to open source and software development for commercial software firms. The introduced method presents a new and structured approach to open source vitality analysis and can help product managers to increasingly implement open source in their products.

Keywords: open source, vitality analysis, health analysis, software ecosystem

1. Introduction

Contemporary, large software products are frequently developed by an entire ecosystem of organizations and open source communities (OSCs) [1]. Such software products base their code on deliverables produced in so called software ecosystems, which span beyond the influence of a single organization [2]. Hereby, open source communities often provide innovative, user demand driven software, free of charge [3]. Implementing open source components into commercial applications has many advantages for software developers, as they can reduce development costs, improve an applications performance and add functionality without the need to invest into according in-house capabilities [4]. Development resources, freed up by open source integration, can be invested in strengthening a software products core capabilities and improving its market competitiveness [5].

For developing and maintaining a software product that depends on open source code, product managers must be able to assess the health of the communities their products depend on. A lack of understanding of a software component's vitality bears not quantifiable risks for the software firm, as well as for customers relying on the product in question. However, such an assessment can be particularly difficult, as open source communities are loosely structured and use collaboration methods that differ greatly from those in commercial software development [6]. Additionally, an open source communities expertise is not centrally structured, less tangible than with commercial products, and bears no obligations for formal technical support.

Currently, little work on the topic of open source vitality exists and no research known to the authors explicitly focuses on the needs of software product managers, evaluating open source for commercial applications. Wahyudin et al. [7] directly addresses open source community health. Based on an evaluation of scientific literature on OS communities and software project monitoring, the authors constructed a software community interaction model, focusing on the three core quality related perspectives of OSCs: The developer community, the user community and the software product. Consequently, the authors constructed two core health indicators that aggregated previously introduced performance variables: Developer contribution and bug service delay. Weiss [8] takes a different approach to assessing OSC popularity and proposes to look at web search engine results. Based on the assumption that a successful OS application will enjoy broad distribution, he proposes four measures of assessing a communities' popularity, based on the number of matching search results. The work of Izquierdo-Cortazar et al. [9] presents some of the more recent OSC vitality related publications. The authors statistically analyze OSC evolvabil-

*Corresponding author. Address: Department of Information and Computing Sciences, University of Utrecht, P.O. Box 80.089, 3508TB Utrecht, The Netherlands. Tel.: +31 (0)30 253 98 96.

Email addresses: i.zaytsev@students.uu.nl (Ivan Zaytsev), s.jansen@cs.uu.nl (Slinger Jansen)

ity and robustness of 1400 FLOSS communities. Hereby, the utilized approach divided the main research question into two sub-questions dealing with: Size and regeneration, as well as interactivity and workload adequacy. Opposite to the efforts of Izquierdo-Cortazar et al., Samoladas et al. [10] evaluate probabilities for FLOSS survival. By mining an open source database, funded by the European Union (FLOSSMetrics), the authors review time series data and attempt to predict the continuation of OS projects. Among all reviewed research , one of the most comprehensive approaches to general OSC vitality is the work of Subramaniam et al. [11]. The paper addressed OSC project success by means of a longitudinal data analysis of OS projects on SourceForge over a time period of 5 years. Based on a literature analysis, the authors constructed a model of open source success measures that was divided in developer interest in the project, user interest in the project and project activity. In addition, the model evaluated interrelations between success factors and introduced a segmentation into time dependent and time independent variables, such as e.g. developer interest or operating system language.

In order to fill the research gap, this article introduces a modular method for OSC vitality analysis, specifically designed to be adjustable to situational requirements of varying communities and project characteristics. The method is built around a pool of method fragments, each aimed at validated OSC vitality indicators and labelled with suitable selection rational. Furthermore, a case study at a large software company with extensive open source expertise was performed. The gained industry insights were used to expand the understanding of OSC vitality indicators, as well as for validation of the presented method.

The following section introduces the utilized research method, covering case study details, as well as a systematic literature review on the topic of OSC and open source related software ecosystem health. Lorem ipsum.....

2. Reserach Method

3. The OSC Health Analysis Method

4. Discussion and Conclusion

References

[1] S. Jansen, A. Finkelstein, S. Brinkkemper, A sense of community: A research agenda for software ecosystems, 31st International Conference on Software Engineering (ICSE 2009) (2009) 2–5.

[2] J. Bosch, From software product lines to software ecosystems, SPLC '09 Proceedings of the 13th International Software Product Line Conference (2009) 1–10.

[3] E. Von Hippel, Learning from open-source software, MIT Sloan management review 42 (2001) 82–86.

[4] J. E. Bessen, Open Source Software: Free Provision Of Complex Public Goods, Social Science Research Network (2001) 57–81.

[5] R. E. Hawkins, The economics of open source software for a competitive firm, NETNOMICS: Economic Research and Electronic Networking 6 (2004) 103–117.

[6] K. Crowston, J. Howison, The social structure of free and open source software development, First Monday 10 (2005) 2–7.

[7] D. Wahyudin, K. Mustofa, A. Schatten, S. Biffl, A. M. Tjoa, Monitoring the "health" status of open source web-engineering projects, International Journal of Web Information Systems - IJWIS 3 (2007) 116–139.

[8] D. Weiss, Measuring success of open source projects using web search engines, Technology (2005) 93–99 TS – BeitraginSammelband.

[9] D. Izquierdo-Cortazar, J. González-Barahona, G. Robles, J. Deprez, V. Auvray, FLOSS Communities: Analyzing Evolvability and Robustness from an Industrial Perspective, Open Source Software New Horizons 319 (2010) 336–341.

[10] I. Samoladas, L. Angelis, I. Stamelos, Survival analysis on the duration of open source projects, Information and Software Technology 52 (2010) 902–922.

[11] C. Subramaniam, R. Sen, M. L. Nelson, Determinants of open source software project success: A longitudinal study, Decision Support Systems 46 (2009) 576–585.