

6-2017

Measuring Adoption of Open Source Software and Practices in Organizations: A Pilot Study

Georg J.P. Link

University of Nebraska at Omaha, glink@unomaha.edu

Jeremiah Gill

University of Nebraska at Omaha, jgill@unomaha.edu

Deepak Khazanchi

University of Nebraska at Omaha, khazanchi@unomaha.edu

Follow this and additional works at: <http://aisel.aisnet.org/mwais2017>

Recommended Citation

Link, Georg J.P.; Gill, Jeremiah; and Khazanchi, Deepak, "Measuring Adoption of Open Source Software and Practices in Organizations: A Pilot Study" (2017). *MWAIS 2017 Proceedings*. 43.

<http://aisel.aisnet.org/mwais2017/43>

This material is brought to you by the Midwest (MWAIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in MWAIS 2017 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Measuring Adoption of Open Source Software and Practices in Organizations: A Pilot Study

Georg J.P. Link 

College of Information Science & Technology
University of Nebraska at Omaha
glink@unomaha.edu

Jeremiah Gill

College of Information Science & Technology
University of Nebraska at Omaha
jgill@unomaha.edu

Deepak Khazanchi

College of Information Science & Technology
University of Nebraska at Omaha
khazanchi@unomaha.edu

ABSTRACT

Modern organizations rely on information systems and technology that utilize open source software and practices to varying degrees. This research-in-progress reports on an effort to measure adoption of open source software and practices (OSS/P) in an organization. For this purpose, we operationalize adoption of OSS/P with a survey instrument based on prior research. The paper contributes to IS research by developing a measure for OSS/P adoption in an organization.

Keywords

Open Source Software Adoption, Open Source Practices Adoption, Survey Research

INTRODUCTION

Many organizations are adopting open source software and practices or OSS/P (Black Duck 2016) because it makes business sense (Ayala et al. 2011; Dedrick and West 2004; Miralles et al. 2006). OSS/P helps share costs, spread risks, increase development speed and aids in innovation, especially when collaborating with competitors on non-differentiating technology (Germonprez et al. 2013; Wesselius 2008). Organizations apply different strategies for adopting and contributing to OSS (Thanasopon 2015). Additionally, organizations use OSS/P for internal software development without having to publicly participate in open communities while retaining control over intellectual property (Ayala et al. 2011; Torkar et al. 2011).

Understanding the level to which an organization adopts OSS/P is important because it allows investigating factors that can foster or impede OSS/P adoption. Several case studies described the complexities involved with organizational adoption of OSS/P (e.g., Fitzgerald et al. 2011). The next logical step is to study antecedents and effects of organizational OSS/P adoption through empirical work and test for generalizability. Culture, for example, might affect the adoption of OSS/P (Pykalainen 2008). This raises the question of how organizational culture affects OSS/P adoption, but to date, there is no quantitative measurement for the level of OSS/P adoption. Before being able to answer such a question, we need to understand how to assess OSS/P adoption in organizations. To address this gap in the research, this paper explores the OSS/P concept and details the initial validation of a measure for OSS/P adoption.

THEORETICAL FOUNDATION

The OSS/P literature is sparse on research on the adoption of OSS/P (Aksulu and Wade 2010). Early works investigated the factors that influence an organization's decision to adopt OSS/P (e.g. Dedrick and West 2004; Miralles et al. 2006). Important decision factors are the availability of long-term support, internal resources, performance, and costs whereas the freedom inherent in OSS/P has little influence (Dedrick and West 2004; Johnston et al. 2013). Decision makers are often unaware of their latent reasons for (not) adopting OSS/P, but general openness towards OSS/P correlates with higher levels of adoption (Miralles et al. 2006). Poba-Nzaou and colleagues (2014) describe how a small manufacturer intuitively adopted a mission critical OSS. In contrast, Mahapatra and colleagues (2015) report on two cases where an early adopting organization abandoned the OSS after several years and a late adopter only used the OSS in non-mission critical operations. Fitzgerald and colleagues (2011) developed a framework for investigating the adoption of OSS adoption and applied it to five case studies. The framework consists of three antecedents (managerial intervention, subjective norms, and facilitating conditions) for secondary adoption of OSS which is expressed in one of five assimilation stages (awareness/interest, evaluation/trial, limited deployment, general deployment, and abandonment). Secondary adoption refers to the stage where the organizational users adopt OSS, which is

contrasted to primary adoption which is the decision to adopt OSS in the organization. For the purpose of this study, we consider secondary adoption that is how and to what degree people in the organization actually adopt OSS.

Extant literature identifies seven ways that organizations adopt OSS/P (Thanasopon 2015): 1) Organizations deploy OSS in their operational environment as end users. 2) Organizations use OSS tools in software development for internal software projects. 3) Organizations integrate OSS components into their software systems. 4) Organizations participate in the development of OSS that is controlled by a community or another company. 5) Organizations release internally developed software under an open source license and create a community around this project. 6) Organizations use open source practices within their organizations. Last, 7) Organizations build business models around OSS. In the context of our research and for secondary adoption, *we consider organizations as having adopted OSS/P when their employees engage with and contribute to open source communities or when internal software development practices are similar to those used by open source communities.*

MEASURING OSS/P

Most studies on the adoption of OSS/P are qualitative (Thanasopon 2015) and no operationalization for the level of OSS/P adoption at the organizational level surfaced in our literature search. To develop a measure for OSS/P adoption, we began by gathering all potential indicators of OSS/P adoption largely based on *a priori* literature (Howison and Crowston 2014; Torkar et al. 2011). The result of this analysis is provided in Appendix A. The indicators intuitively cluster into three categories: involvement with open source communities, having transparent processes, and adopting OSS practices. Using Babbie's (1990) recommendation to convert concepts into survey questions, the authors went through several iterations for writing questions, discussing them with experts, and rewriting them for clarification. Since we focus on the secondary adoption (Fitzgerald et al. 2011), the questions were written for employees in the organization. The instrument was reviewed by an outside open source scholar and improved based on the feedback. We conducted a pilot study by sending out a web-based survey to a US-based global provider of communication and network infrastructure services. Research participants for our study are members of the organization's IT department. The 11-item survey (refer Appendix A) is on a Likert-type scale (1 to 5) with the options for "N/A" or "Don't know" responses. Responses with four or more missing data were removed. Four participants agreed to be interviewed and give detailed feedback on the way the questions were perceived.

RESULTS

We collected 23 valid surveys. A summary of respondent demographics is included in Table 1 below.

Age (years)		Sex	
< 37	26%	Male	83%
37-52	61%	Female	17%
> 52	13%		
Experience (years)		Title/Position	
< 5	39%	Engineer	74%
5-9	26%	Manager	13%
9-13	13%	Other	13%
> 13	22%		

Table 1. Participant Demographics

Even with the limited responses, the OSS/P adoption instrument is highly reliable (12 items, Cronbach $\alpha = .817$; mean inter-item correlation: .293). The reliability of the instrument increases when dropping the question about personal level of participation (11 items, $\alpha = .832$), possibly because it is not an organizational level question. The involvement subscale without the question about personal participation consisting of 5 items had an α of .784, the transparency subscale consisting of 2 items had an α of .771, and the practices subscale consisting of 4 items had an α of .458.

The interviews with four participants provided insights for improving the questions, especially in terms of language. For example, the term 'change stack' did not make sense to the participants since they do not use that vocabulary in their work. This created ambiguity since 'change stack' was interpreted as the tools used in open source development or a system to track

what open source was in use in the organization. Both interpretations deviate from the intended meaning of tracking changes made to an open source software so as to be able to apply the same change to future releases. In another example, a participant pointed out that the differentiation between synchronous and asynchronous tools for decision making did not make sense to him. Apparently, a meeting can be split up into work groups to achieve synchronicity in the work. Conversely, emails are answered in real time and can be seen as synchronous communication in the organization.

* To what degree does your organization maintain small software development tasks that you can pick up when you have time and complete independently?

1: Tasks are too large or complex that I cannot just pick one up when I have some spare time

2

3: A few tasks I can do when I have a slow time

4

5: I can pick almost any task and complete it independently in a short amount of time

N/A

Don't know

Task size

Description/Example: Tasks can be small and independent where one person can go choose any task and complete it. The alternative are large tasks that require a long time to complete and potentially need coordination with other developers.

Figure 1 Example Question from Online Survey.

Online survey providers allow for additional explanation of a question, the blue text at the bottom of Figure 1. During the follow-up interview we asked respondents to comment on the additional explanation. Apparently, respondents focused exclusively on the question and ignored the additional information. After reading the information during the interview, respondents commented that it was helpful to clarify the question and eliminated uncertainty. Based on the feedback we received, we improved the language and clarity of the questions.

CONCLUSION

This paper briefly explains the development of an instrument to measure OSS/P adoption in organizations and demonstrates using pilot data that the instrument is potentially reliable. We interviewed four participants who helped us identify ambiguous questions and refine the instrument. We plan to validate the improved instrument with a larger sample size and include organizations from other industries. This will be done through online surveys, just as described for the pilot study. Once we establish that the validity and reliability of the instrument we will use it in other research projects. For example, we will test our notion that organizational culture may influence OSS/P adoption. This is because we know that, among other things, organizational culture has an effect on performance (Kotter and Heskett 1992), employee retention (Trübswetter et al. 2016), information systems success (Bradley et al. 2006), and technology adoption and diffusion (Leidner and Kayworth 2006). However, how organizational culture affects OSS/P adoption is not well understood (Pykalainen 2008).

ORCID

Georg J.P. Link  <https://orcid.org/0000-0001-6769-7867>

REFERENCES

1. Aksulu, A., and Wade, M. 2010. "A Comprehensive Review and Synthesis of Open Source Research," *Journal of the Association for Information Systems*, (11:11).
2. Ayala, C. P., Cruzes, D., Hauge, O., and Conradi, R. 2011. "Five Facts on the Adoption of Open Source Software," *IEEE Software*, (28:2), pp. 95–99 (doi: 10.1109/MS.2011.32).
3. Babbie, E. R. 1990. *Survey research methods*, (2nd ed.), Belmont, Calif: Wadsworth Pub. Co.
4. Black Duck. 2016. "The tenth annual future of open source survey," *Black Duck Software* (available at <https://www.blackducksoftware.com/2016-future-of-open-source>; retrieved October 11, 2016).

5. Bradley, R. V., Pridmore, J. L., and Byrd, T. A. 2006. "Information systems success in the context of different corporate cultural types: an empirical investigation," *Journal of Management Information Systems*, (23:2), pp. 267–294 (doi: 10.2753/MIS0742-1222230211).
6. Dedrick, J., and West, J. 2004. "An exploratory study into open source platform adoption," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, IEEE.
7. Fitzgerald, B., Kesan, J. P., Russo, B., Shaikh, M., and Succi, G. 2011. *Adopting open source software: a practical guide*, Cambridge, Massachusetts: MIT Press.
8. Germonprez, M., Allen, J. P., Warner, B., Hill, J., and McClements, G. 2013. "Open source communities of competitors," *ACM Interactions*, (20:6), pp. 54–59 (doi: 10.1145/2527191).
9. Howison, J., and Crowston, K. 2014. "Collaboration through open superposition: A theory of the open source way," *MIS Quarterly*, (38:1), pp. 29-A9.
10. Johnston, K., Begg, S., and Tanner, M. 2013. "Exploring the factors influencing the adoption of Open Source Software in Western Cape schools," *International Journal of Education and Development using Information and Communication Technology*, (9:2), p. 64.
11. Kotter, J. P., and Heskett, J. L. 1992. *Corporate culture and performance*, New York : Toronto : New York: Free Press ; Maxwell Macmillan Canada ; Maxwell Macmillan International.
12. Leidner, D. E., and Kayworth, T. 2006. "Review: A review of culture in information systems research: Toward a theory of information technology culture conflict," *MIS Quarterly*, (30:2), pp. 357–399.
13. Mahapatra, R., Manzar, R., and Bhadauria, V. S. 2015. "Adoption and use of open source infrastructure software by large corporations: The case of MySQL," *Journal of Database Management (JDM)*, (26:4), pp. 1–17.
14. Miralles, F., Sieber, S., and Valor, J. 2006. "An exploratory framework for assessing open source software adoption," *Systèmes d'Information et Management*, (11:1).
15. Poba-Nzaou, P., Raymond, L., and Fabi, B. 2014. "Risk of adopting mission-critical OSS applications: an interpretive case study," *International Journal of Operations & Production Management*, (34:4), pp. 477–512.
16. Pykalainen, T. 2008. "Adaption of Linux SSL servers across cultures," *First Monday*, (13:12).
17. Thanasopon, B. 2015. "Open Source Software (OSS) adoption in commercial firms: A literature review," *KMITL Information Technology Journal*, (4:1).
18. Torkar, R., Minoves, P., and Garrigós, J. 2011. "Adopting free/libre/open source software practices, techniques and methods for industrial use," *Journal of the Association for Information Systems*, (12:1).
19. Trübswetter, A., Genz, K., Hochfeld, K., and Schraudner, M. 2016. "Corporate culture matters – what kinds of workplaces appeal to highly skilled engineers?," *International Journal of Gender, Science and Technology*, (8:1), pp. 46–66.
20. Wesselius, J. 2008. "The Bazaar inside the Cathedral: Business models for internal markets," *IEEE Software*, (25:3), pp. 60–66 (doi: 10.1109/MS.2008.79).

APPENDIX A: OPEN SOURCE SOFTWARE AND PRACTICES ADOPTION INSTRUMENT (OSSPAI)

Category	Question	Description / Example	Scale: 1-5, N/A, Don't know
Involvement	What is your personal level of participation in open source project(s)?	You are commenting on issues, discussing on the mailing list, or contributing patches to add features or fix bugs.	1: No participation, I am not involved 3: Moderate participation, I am involved but sporadically or infrequently 5: Significant participation, I contribute code, documentation, or testing to a project on a regular basis
Involvement	To what degree does your organization allow employee participation in open source projects?	Employees are participating in the open source community by commenting on issues, discussing on the mailing list, or contributing patches to add features or fix bugs.	1: Not at all. My organization does not allow employees to participate in open source communities, even during off-hours 2: My organization does not allow employees to participate in open source communities while at work 3: Moderate participation, my organization does not expect us to participate in open source projects but we can if it helps our job 4: We are encouraged to participate in open source communities as part of our job 5: Significant degree. My organization dedicates employees to participate in open source communities as part of their job
Involvement	To what degree does your organization maintain a change stack for open source software it uses?	When developing product software, the tendency is often to use open source components, and then apply internally-developed patches for performance, custom features, and so on. Over time, this out-of-band code accumulates and is maintained in change stacks. Change stacks are applied to releases to maintain the custom features. If left unchecked it can result in an unwieldy code base that is complicated to port forward to new products at best. In the worst case, the sheer complexity and communal knowledge required to maintain such patches can open up surprising and unexpected attack vectors.	1: Not at all. My organization does not modify open source software 3: Significant degree. My organization maintains large change stack(s) and applies them to every new release of an open source project 5: Moderate degree. My organization tries to minimize change stack(s) by getting as many changes accepted upstream in the original open source project as possible
Involvement	To what degree does your organization contribute to open source projects?	The organization might contribute to open source projects for a variety of reasons, e.g. to commit changes upstream, to maintain a critical library, or for staying current with innovative projects. Following an open source project means, for example, to stay up to date on releases, to subscribe to a mailing list, or following it on social media.	1: Not at all 3: Moderate degree. My organization follows open source projects but does not contribute regularly 5: Significant degree. My organization regularly contributes to open source projects

Involvement	To what degree does your organization release software under an open source license and foster a community?	Tools and other software developed internal to the organization have been licensed under an open source license and the organization fosters a community of outsiders to participate in the development of the software.	1: Not at all. My organization does not release any source code of software under an open source license 3: Moderate degree. My organization releases the source code under an open source license but is not actively fostering a community of outside contributors 5: Significant degree. My organization fosters healthy and sustainable open source communities around software it releases
Involvement	How long has your organization worked with open source?		(different answer options): Less than 1 year 1 to 3 years 3 to 5 years More than 5 years
Transparency	To what degree does your organization value transparency of internal software products?	The organization has a repository of internally developed tools and other software that anyone from within the organization can access to get the latest version and find out when new releases are made.	1: Not at all. In my organization, only the people developing or deploying software can see when a new version is released 3: Moderate degree. Some teams can see when internal software is released 5: Significant degree. Anyone at the organization can see when software is released
Transparency	To what degree does your organization value transparency in the internal software development process?	Anyone has access to bug-tracker software (e.g. Jira, Bugzilla, Redmine), can see the current state of issues, and can download the work-in-progress repositories. The documentation is updated as features change, and everyone can see the latest version of the documentation.	1: Not at all. Only the development team knows the status of their projects 3: Moderate: Some teams have access to the status of projects 5: Significant degree. Everyone in the organization can find out what the status of the software development process is
Practice	To what degree does your organization use asynchronous tools, such as emails, to make software development related decisions?	Emails, bug-trackers, wikis and other tools allow users to interact with each other when the other users are not online at the same time (asynchronous). There is no need to schedule a meeting time to make a decision in person or over Skype (synchronous).	1: Not at all. We always use meetings or other synchronous decision-making 3: To some degree. We make some decisions using asynchronous tools but also rely on meetings and other synchronous decision-making 5: To a significant degree. We make all decisions asynchronously with very few scheduled meetings
Practices	To what degree does your organization value quality and stable products compared to quick solutions?	Employees are given the time to refactor code, develop an optimal solution, and are not pressured to deliver quick fixes that complicate future maintenance of the code base.	1: My organization values mostly quick solutions, even if we have to fix bugs later 3: My organization values a balance of quality and quick solutions 5: My organization values mostly quality and stability, and we are given sufficient time to develop elegant solutions that are bug-free

Practices	To what degree are software development activities, such as code review, recorded and available through an archive?	The organization maintains an archive of code reviews and comments made by the reviewers as well as whether changes to the code were made consequently.	1: Not at all. Records are not available 3: To some degree. Records are temporarily available but cannot be accessed after some time 5: To a significant degree. Records are permanently available through an archive
Practices	To what degree does your organization maintain small software development tasks that you can pick up when you have time and complete independently?	Tasks can be small and independent where one person can choose any task and complete it. The alternative are large tasks that require a long time to complete and potentially need coordination with other developers.	1: Tasks are too large or complex that I cannot just pick one up when I have some spare time 3: A few tasks I can do when I have a slow time 5: I can pick almost any task and complete it independently in a short amount of time