Steinmacher et al. Journal of Internet Services and Applications (2017) 8:17
DOI 10.1186/s13174-017-0069-9

Journal of Internet Services and Applications

EDITORIAL Open Access

Free and open source software development: the end of the teenage years

Igor Steinmacher[1,2*], Gregorio Robles[3], Brian Fitzgerald[4] and Anthony Wasserman[5]

1 Introduction

This thematic series of the *Journal of Internet Services and Applications (JISA)* presents a collection of articles around the broad topic of the development of Free and Open Source Software (FOSS). This series illustrates the diversity of topics that are related to FOSS – we received submissions from many different areas; for example, dealing with communication and coordination within FOSS communities, licensing, adoption of Kanban in FOSS settings, and the use of an open model in the editorship of standards. Moreover, the studies presented were conducted using data collected from different sources, including software repositories, mailing lists, interviews, questionnaires and field notes, and were analyzed using quantitative and qualitative methods, including case studies and action research methods.

The FOSS model has become an important driving force in today's software development environment, resulting in many prominent projects that are used extensively through the entire development stack, from kernels to sophisticated end-user applications. Even startups and commercial projects are increasingly participating in FOSS [1]. The 10th Annual Future of Open Source Survey [2] showed that 65% of the surveyed companies leverage FOSS to speed up application development, and 55% leverage FOSS for production infrastructure. On the other hand, the movement has introduced new challenges [3–7], especially in relation to openness and the innovative nature of this development model.

The openness and success of the FOSS model is a good topic for research. Researchers from different areas and backgrounds are able to analyze publicly available data to better understand how software is developed [8, 9], learn from it [10, 11], and to propose ways to improve FOSS processes [12–14]. The available data range from technical artifacts, to messages exchanged via mailing lists, forums and chats, and discussions on tasks. All these factors attracted and enabled studies related to management [15–17], software architecture and evolution [9, 18–20], software testing [21], human and social aspects of software engineering [22–25], software engineering education [26–29], etc.

One issue worth highlighting is that FOSS projects are open collaboration communities, in which social aspects are important [30]. Therefore, analyzing the social structure of FOSS projects and behavior within communities has attracted attention from many diverse researcher groups. For example, there is a huge body of knowledge on the motivation of volunteers that joined the workforce of FOSS projects [31–35]. The developer-joining process also drew the attention of researchers from different areas attempting to understand how developers join [36–39], evolve [40, 41], and become central [42, 43] in a FOSS community.

The FOSS model has spread well beyond software, and inspired different initiatives in different domains [44–51] such as open innovation [44, 51], open hardware [46, 49, 52], open politics [47, 48], open content (e.g., Wikipedia, OpenStreetMap) [50, 53, 54] and open education resources [55, 56].

2 Papers included in this thematic series

The set of papers selected for this thematic series covers different research approaches, providing a broad overview of the area, with its tools, platforms, and research methods. Next, we briefly summarise the main aspects of the individual papers accepted for publication in this special issue.

As mentioned before, FOSS projects are, in general, open collaboration communities, in which social relationships are important. Social relationships become more complicated when it is necessary to orchestrate the efforts to communicate and coordinate effectively throughout an ecosystem of interrelated software products to achieve a specific goal. In the paper "Herding

* Correspondence: igorfs@utfpr.edu.br
[1]School of Informatics, Computing, and Cyber Systems – Northern Arizona University, Flagstaff, AZ, USA
[2]Department of Computing, Federal University of Technology, Campo Mourão, PR, Brazil
Full list of author information is available at the end of the article

cats in a FOSS ecosystem: a tale of communication and coordination for release management" [57], Germán Poo-Camaño and his colleagues conducted a case study to better understand how communication and coordination take place in release management of the GNOME ecosystem. They quantitatively and qualitatively analyzed two and a half years of projects' mailing lists and interviewed developers. Poo-Caamaño and colleagues found that, for GNOME, mailing lists are still the main communication channel used for the release management team, which also make strong use of an IRC channel to conduct synchronous meeting and to discuss with developers. They use the mailing list to discuss different things, including requests for comments/approval, release announcement and to send schedule reminders. In addition, the authors identified the main responsibilities and activities conducted by the GNOME release team, and the challenges to manage releases in FOSS ecosystems. They end up bringing seven valuable lessons learned from their results.

Social relationships are also explored in the paper "Core-periphery communication and the success of free/libre open source software projects." [58] In this paper, Crowston and Shamshurin bring an study on how community interactions relates to project health and success in Apache projects. The authors are explicitly interested in how volume and content of messages sent via mailing list by core and periphery member can be related to success of projects. Their results indicate that successful projects have a larger volume of communication - suggesting an active community. They also found that, although both core and periphery members place a high volume of messages, core members communicate more than those from the periphery. Regarding the content of the messages, they found no association between project success and the group feeling of ownership of the project.

Other than (and adding to) the community and their social relationship, another central element of FOSS is licensing. Licenses grant specific permissions that regulate how source code can be further used. FOSS licenses make it easy for others to contribute to a project without having to seek special permission. In the paper "Changes in free and open source software licenses: managerial interventions and variations on project attractiveness," [59] Carlos Denner Jr. analyzes the relationship between changes in licensing terms and projects' attractiveness. By analyzing 756 FOSS projects that changed their license, Denner observed 35 different variation patterns. His results indicate that the legal terms in the license are associated with project attractiveness, however variations in attractiveness after a license intervention are not symmetrical.

In the paper "Can FOSS projects benefit from integrating Kanban" [60], Annemarie Harzl explores how the use of Kanban is perceived by a FOSS community mainly kept by students. By means of three action-research cycles, the author found that Kanban in these specific settings are beneficial, both for managing the tasks and as a learning opportunity for the students. The paper shows an interesting perspective on how a FOSS project can be maintained in hybrid settings, considering students and applying the coach agile role to manage the project. It also shows, ultimately, that adopting FOSS as a learning environment, by putting students to work on real projects, can benefit both project and students' learning process.

A technical aspect of FOSS development is addressed by the paper "Concurrency bugs in open source software: a case study" [61] by Sara Abbaspour Asadollah and her colleagues. In this paper, the authors are interested in better understanding the differences between concurrency-related bugs and other bugs by analyzing bug reports from five Apache projects. They found that the main type of concurrency bugs are related to data race condition, but suspension, deadlock and order violation also appear with some frequency. An interesting finding is that concurrency bugs are more complex than other bugs. The paper reports that, in general, concurrency bugs require longer fixing times and that around 50% of the unreproducible bugs are concurrency bugs. The authors claim that the results help software developers to understand concurrency bugs, estimate the most time-consuming ones, and prioritizing them to speed up the debugging and bug-fixing processes.

Showing how the open model is spread, the paper from Jonas Gamalielsson and Björn Lundell, "On organisational involvement and collaboration in W3C standards through editorship" [62], brings a characterization on how organizations contribute to W3C open standards related to software development. The authors analyzed all the contributions to W3C standards and found that one standardisation organisation and larger enterprises (mainly from US) are dominating involvement in the open standards editorship. However, they also observed a large number of small and medium enterprises, from a wide range of countries, involved in a relatively large number of standards, highlighting the openness of the process. In addition, by analyzing the social network of organizations and countries, it was possible to notice that there is a high level of collaboration between organizations and countries, with larger organizations and US playing a central role. The paper shows the openness of the standards editorship, bringing insights and implications for W3C and for the organizations that want to take place on this endeavour, or engage a collaboration.

3 Paper selection

There were two independent cycles of submissions and the papers were published after acceptance and the final

version received from the authors. We submitted an open call for submissions, and invited the three best papers from the Open Source Systems conference (OSS 2016) in Gothenburg, Sweden –one of the special issue editors was one of the OSS 2016 program co-chairs– to submit extended versions of their papers to this series. Each submitted manuscript went through several revisions before the final acceptance. We invited a number of leading experts in the area to form an initial editorial committee, and additional reviewers had been invited on demand in order to provide valuable feedback for the authors and review different aspects of the papers. All manuscripts were reviewed by at least three reviewers. We received a total of 10 submissions, from which we accepted six for publication. All accepted papers are extensions of previously published papers. Five of them extending previous work from the Open Source Systems Conference. The papers were reviewed by a total of 31 reviewers. The names of the editorial committee and reviewers are listed on the acknowledgements of this editorial.

4 Concluding remarks

This is clearly not the last word on free and open source software, but as Eric Raymond has aptly characterised it – last words are about dead things and open source software is quite lustily alive. Although the history of free and open source software is as old as software itself, there is no doubt that considerable impetus was afforded to the topic when the term open source was coined in 1998. Given that almost 20 years have elapsed since then, it is interesting to reflect on how much the field has evolved in that period. This is very much in evidence in the range of topics and issues covered in the papers submitted to this thematic series.

Acknowledgements

We thank all the authors, reviewers, editors-in-chief, and staff for the great work, which supported this thematic series on this important topic. In particular, we thank all the reviewers:

Andrea Capiluppi (Brunel University, UK), Ann Barcomb (University of Limerick, Ireland), Anna Filippova (CMU, USA), Aron Lindberg (Stevens Institute of Technology, USA), Christoph Treude (University of Adelaide, Australia), Daniel Germán (UVic, Canada), Erik Trainer (CMU, USA), Fernando Figueira (UFRN, Brazil), Filipe Saraiva (UFPI, Brazil), Gustavo Ansaldi Oliva (Queen's University, Canada), Gustavo Pinto (UFPA, Brazil), Habtom Gidey (University of Limerick, Ireland), Harry Halpin (W3C, UK), Igor Scaliante Wiese (UTFPR, Brazil), Shobha Chengalur-Smith (University at Albany, USA), Irum Inayat (National University of Computer and Emerging Sciences, Pakistan), Jason Tsay (CMU, USA), Johan Linåker (Lund University, Sweden), Jose Teixeira (University of Turku, Finland), Josiane Kroll (PUCRS, Brazil), Kelly Blincoe (University of Auckland, New Zealand), Kevin Crowston (Syracuse University, USA), Marcelo Zanetti (UFSM, Brazil), Marco Aurélio Graciotto Silva (UTFPR, Brazil), Marco Ortu (University of Cagliari, Italy), Mauricio Aniche (TU Delft, Nederlands), Paulo Barbosa (UEPB, Brazil), Shane McIntosh (McGill University, Canada), Sven Apel (University of Passau, Germany), Tiago Silva da Silva (UNIFESP, Brazil), Zhendong Wu (National University of Defense Technology, China). The voluntary work of these researchers was crucial for this thematic series. This work was supported by the Science Foundation Ireland grant 13/RC/2094.

Publisher's Note

Author details

[1]School of Informatics, Computing, and Cyber Systems – Northern Arizona University, Flagstaff, AZ, USA. [2]Department of Computing, Federal University of Technology, Campo Mourão, PR, Brazil. [3]Sistemas Telematicos y Computacion, Universidad Rey Juan Carlos, Madrid, Spain. [4]Lero: the Irish Software Research Centre - University of Limerick, Limerick, Ireland. [5]Carnegie Mellon University, Silicon Valley, Moffett Field, CA, USA.

Received: 13 November 2017 Accepted: 13 November 2017
Published online: 05 December 2017

References

1. Anthes, G. 2016. Open source software no longer optional. Commun ACM. 59, 8, 15–17. DOI:https://doi.org/https://doi.org/10.1145/2949684.
2. BlackDuck 2016. The Tenth Annual Future of Open Source Survey. Available at: https://www.blackducksoftware.com/2016-future-of-open-source. Accessed 08 Nov 2016.
3. Adams, B., Kavanagh, R., Hassan, A.E. and German, D.M. 2016. An empirical study of integration activities in distributions of open source software. Empir Softw Eng. 21, 3, 960–1001. DOI:https://doi.org/https://doi.org/10.1007/s10664-015-9371-y.
4. Krafft MF, Stol K-J, Fitzgerald B. How do free/open source developers pick their tools? A Delphi study of the Debian project. In: Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on (2016); 2016. p. 232–41.
5. Michlmayr M, Fitzgerald B, Stol K-J. Why and how should open source projects adopt time-based releases? IEEE Softw. 2015;32(2):55–63.
6. Poo-Caamaño G, Singer L, Knauss E, German DM. Herding cats: a case study of release management in an open collaboration ecosystem. In: Crowston K, Hammouda I, Lundell B, Robles G, Gamalielsson J, Lindman J, editors. Open Source Systems: Integrating Communities. OSS 2016. IFIP Advances in Information and Communication Technology, vol 472. Cham: Springer; 2016.
7. Steinmacher I, Silva MAG, Gerosa MA. Barriers faced by newcomers to open source projects: a systematic review. In: Corral L, Sillitti A, Succi G, Vlasenko J, Wasserman AI, editors. Open Source Software: Mobile Open Source Technologies. OSS 2014. IFIP Advances in Information and Communication Technology, vol 427. Berlin: Springer; 2014.
8. Cerulo L. On the use of process trails to understand software development. Reverse engineering, 2006. WCRE'06. 13th working conference on (2006); 2006. p. 303–4.
9. D'Ambros M, Gall H, Lanza M, Pinzger M. In: Mens T, Demeyer S, editors. Analysing software repositories to understand software evolution. Software Evolution. Heidelberg: Springer Berlin; 2008. p. 37–67.
10. Hauge Ø, Sørensen C-F, Conradi R. In: Russo B, Damiani E, Hissam S, Lundell B, Succi G, editors. Adoption of open source in the software industry. Open source development, communities and quality: IFIP 20th world computer congress, working group 2.3 on open source software, September 7–10, 2008, Milano, Italy. US: Springer; 2008. p. 211–21.
11. Stol K-J, Fitzgerald B. Inner source–adopting open source development practices in organizations: a tutorial. IEEE Softw. 2015;32(4):60–7.
12. Anvik, J. and Murphy, G.C. 2011. Reducing the effort of bug report triage: recommenders for development-oriented decisions. ACM Transactions on Software Engineering and Methodology (TOSEM). 20, 3 10.
13. Hassan AE. Mining software repositories to assist developers and support managers. In: 22nd IEEE International Conference on Software Maintenance (sep. 2006); 2006. p. 339–42.
14. Steinmacher I, Conte TU, Treude C, Gerosa MA. Overcoming open source project entry barriers with a portal for newcomers. In: 38th International Conference on Software Engineering (2016); 2016. p. 273–84.
15. von Hippel E, von Krogh G. Open source software and the "private-collective" innovation model: issues for organization science. Organ Sci. 2003;14(2):209–23.
16. Howison J, Crowston K. Collaboration through open superposition: a theory of the open source way. MIS Q. 2014;38(1)
17. Iskoujina Z, Roberts J. Knowledge sharing in open source software communities: motivations and management. J Knowl Manag. 2015; 19(4):791–813.

18. Lehman MM. Laws of software evolution revisited. In: Montangero C, editor. Software Process Technology. EWSPT 1996. Lecture Notes in Computer Science, vol 1149. Berlin: Springer; 1996.
19. Nakagawa EY, de Sousa EPM, de Brito Murata K, de Faria Andery G, Morelli LB, Maldonado JC. Software architecture relevance in open source software evolution: a case study. 32nd Annual IEEE International Computer Software and Applications; 2008. p. 1234–9.
20. Oliva GA, Steinmacher I, Wiese I, Gerosa MA. What can commit metadata tell us about design degradation?. In Proceedings of the 2013 International Workshop on Principles of Software Evolution (IWPSE 2013). New York: ACM; 2013. pp. 18-27. http://doi.acm.org/10.1145/2501543.2501547
21. Aniche MF, Oliva GA, Gerosa MA. What do the asserts in a unit test tell us about code quality? A study on open source and industrial projects. 2013 17th European Conference on Software Maintenance and Reengineering. Washington, DC: IEEE CS; 2013. p. 111–20.
22. Sarma A, Maccherone L, Wagstrom P, Herbsleb J. Tesseract: interactive visual exploration of socio-technical relationships in software development. In: 31st International Conference on Software Engineering (Washington, DC, USA, 2009); 2009. p. 23–33.
23. Steinmacher I, Conte T, Gerosa MA, Redmiles DF. Social barriers faced by newcomers placing their first contribution in open source software projects. 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (new York, NY, USA, Feb. 2015); 2015. p. 1–13.
24. Steinmacher I, Wiese IS, Chaves AP, Gerosa MA. Why do newcomers abandon open source software projects? 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (2013); 2013. p. 25–32.
25. Storey M-A, Treude C, van Deursen A, Cheng L-T. The impact of social media on software engineering practices and tools. FSE/SDP Workshop on Future of Software Engineering Research (New York, NY, USA, 2010); 2010. p. 359–64.
26. Van Deursen A, Aniche M, Aué J, Slag R, De Jong M, Nederlof A, Bouwers E. A collaborative approach to teaching software architecture. Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education; 2017. p. 591–6.
27. Ellis HJC, Chua M, Hislop GW, Purcell M, Dziallas S. Towards a model of faculty development for FOSS in education. In: 2013 26th International Conference on Software Engineering Education and Training (CSEE&T) (may 2013); 2013. p. 269–73.
28. Pinto G, Figueira Filho F, Steinmacher I, Gerosa MA. Training software engineers using open-source software: the professors' perspective. The 30th IEEE Conference on Software Engineering Education and Training; 2017. p. 1–5.
29. Sarma A, Gerosa MA, Steinmacher I, Leano R. Training the future workforce through task Curation in an OSS ecosystem. 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (New York, NY, USA, 2016); 2016. p. 932–5.
30. Forte, A. and Lampe, C. 2013. Defining, understanding, and supporting open collaboration: lessons from the literature. Am Behav Sci. 57, 5, 535–547. DOI:https://doi.org/https://doi.org/10.1177/0002764212469362.
31. Hannebauer C, Gruhn V. Motivation of newcomers to FLOSS projects. 12th international symposium on open collaboration (2016); 2016. p. 1.
32. Hars A, Ou S. Working for free? Motivations of participating in open source projects. In: 34th Annual Hawaii International Conference on System Sciences (2001); 2001. p. 1–9.
33. von Krogh G, Haefliger S, Spaeth S, Wallin MW. Carrots and rainbows: motivation and social practice in open source software development. MIS Q. 2012;36(2):649–76.
34. Lakhani KR, Wolf RG. Perspectives on free and open source software. In: Feller J, editor. Perspectives on Free and Open Source Software. Cambridge: The MIT Press; 2005. p. 1–22.
35. Oreg S, Nov O. Exploring motivations for contributing to open source initiatives: the roles of contribution context and personal values. Comput Hum Behav. 2008;24(5):2055–73.
36. Fagerholm, F., Johnson, P., Guinea, A.S., Borenstein, J. and Münch, J. 2014. Onboarding in open source projects. IEEE Softw. 31, 6, 54–61. DOI:https://doi.org/https://doi.org/10.1109/MS.2014.107.
37. Herraiz I, Robles G, Amor JJ, Romera T, Barahona JMG, Carlos J. The processes of joining in global distributed software projects. 2006 international workshop on global software development for the practitioners (2006); 2006. p. 27–33.
38. von Krogh G, von Hippel E. Editorial: special issue on open source software development. Res Policy. 2003;32(7):1149–57.
39. Steinmacher, I., Gerosa, M.A. and Redmiles, D. 2014. Attracting, Onboarding, and retaining newcomer developers in open source software projects. Workshop on Global Software Development in a CSCW Perspective.
40. Jergensen C, Sarma A, Wagstrom P. The onion patch: migration in open source ecosystems. 19th ACM SIGSOFT symposium and the 13th European conf. On foundations of software engineering (2011); 2011. p. 70–80.
41. Nakakoji K, Yamamoto Y, Nishinaka Y, Kishida K, Ye Y. Evolution patterns of open-source software systems and communities. Proceedings of the International Workshop on Principles of Software Evolution (IWPSE '02). New York: ACM. pp. 76-85. http://dx.doi.org/10.1145/512035.512055.
42. Schilling A, Laumer S, Weitzel T. Who will remain? An evaluation of actual person-job and person-team fit to predict developer retention in FLOSS projects. In: 2012 45th Hawaii International Conference on System Sciences (Washington, DC, USA, 2012); 2012. p. 3446–55.
43. Zhou, M. and Mockus, A. 2015. Who will stay in the FLOSS community? Modelling Participant's initial behaviour. IEEE Transactions on Software Engineering. 41, 1, 82–99. DOI:https://doi.org/http://doi.ieeecomputersociety.org/https://doi.org/10.1109/TSE.2014.2349496.
44. Bican, P.M., Guderian, C.C. and Ringbeck, A. Managing knowledge in open innovation processes: an intellectual property perspective. J Knowl Manag 2017 DOI:https://doi.org/https://doi.org/10.1108/JKM-11-2016-0509.
45. Coffin J. An analysis of open source principles in diverse collaborative communities. First Monday. 2006;11(6). http://dx.doi.org/10.5210/fm.v11i6.1342.
46. Dollar A, Mondada F, Rodriguez A, Metta G. Open-source and widely disseminated robot hardware [from the guest editors]. IEEE Robot Automation Mag. 2017;24(1):30–1.
47. Hindman M. Open-source politics reconsidered: emerging patterns in online political participation. Governance and information technology: from electronic government to information government; 2007. p. 183–207.
48. Husted E, Plesner U. Spaces of open-source politics: physical and digital conditions for political organization. Organization. 2017;24(5):648–70.
49. Pearce JM. Laboratory equipment: cut costs with open-source hardware. Nature. 2014;505(7485):618.
50. Royal C, Kapila D. What's on Wikipedia, and what's not...? Assessing completeness of information. Soc Sci Comput Rev. 2009;27(1):138–48.
51. West J, Bogers M. Leveraging external sources of innovation: a review of research on open innovation. J Prod Innov Manag. 2014;31(4):814–31.
52. Pearce JM. Emerging business models for open source hardware. J Open Hardware. 2017;1(1)
53. Haklay M, Weber P. Openstreetmap: User-generated street maps. IEEE Pervasive Computing. 2008;7(4):12–8.
54. Halfaker A, Geiger RS, Morgan J, Riedl J. The rise and decline of an open collaboration system: how Wikipedia's reaction to sudden popularity is causing its decline. Am Behav Sci. 2013;57(5):664–88.
55. Hylén J. Open educational resources: opportunities and challenges. Proc Open Educa. 2006:49–63.
56. Shelton BE, Duffin J, Wang Y, Ball J. Linking open course wares and open education resources: creating an effective search and recommendation system. Procedia Comput Sci. 2010;1(2):2865–70.
57. Poo-Caamaño, G., Knauss, E., Singer, L. and German, D.M. 2017. Herding cats in a FOSS ecosystem: a tale of communication and coordination for release management. J Internet Serv Appl. 8, 1, 12. DOI:https://doi.org/https://doi.org/10.1186/s13174-017-0063-2.
58. Crowston, K. and Shamshurin, I. 2017. Core-periphery communication and the success of free/libre open source software projects. J Internet Serv Appl. 8, 1, 10. DOI:https://doi.org/https://doi.org/10.1186/s13174-017-0061-4.
59. Santos CD. Changes in free and open source software licenses: managerial interventions and variations on project attractiveness. J Internet Serv Appl. 2017;8(1):11. https://doi.org/10.1186/s13174-017-0062-3.
60. Harzl, A. 2017. Can FOSS projects benefit from integrating Kanban: a case study. J Internet Serv Appl. 8, 1, 7. DOI:https://doi.org/https://doi.org/10.1186/s13174-017-0058-z.
61. Abbaspour Asadollah, S., Sundmark, D., Eldh, S. and Hansson, H. 2017. Concurrency bugs in open source software: a case study. J Internet Serv Appl. 8, 1, 4. DOI:https://doi.org/https://doi.org/10.1186/s13174-017-0055-2.
62. Gamalielsson, J. and Lundell, B. 2017. On organisational involvement and collaboration in W3C standards through editorship. J Internet Serv Appl. 8, 1, 5. DOI:https://doi.org/https://doi.org/10.1186/s13174-017-0056-1.