



An author-based review of the Journal of Open Research Software

DANIEL GRAZIOTIN¹

1. Free University of Bozen-Bolzano

ABSTRACT

The Journal of Open Research Software (JORS) is an open access journal, which publishes peer reviewed software papers. Software papers describe open source software for research with high reuse potential. The authors publishing in the journal are awarded for opening up software with a peer reviewed journal article. This article is an author-based review of JORS and an experience report of the submission process of one now published paper there.

READ REVIEWS

WRITE A REVIEW

CORRESPONDENCE:
daniel.graziotin@unibz.it

DATE RECEIVED:
June 11, 2015

DOI:
10.15200/winn.140326.62772

ARCHIVED:
June 20, 2014

KEYWORDS:
software implementation, open
source, reusability, author
experience

CITATION:
Daniel Graziotin, An author-
based review of the Journal of
Open Research Software, *The
Winnower* 2:e140326.62772 ,
2014 , DOI:
10.15200/winn.140326.62772

© Graziotin This article is
distributed under the terms of
the [Creative Commons
Attribution 4.0 International
License](#), which permits
unrestricted use, distribution,
and redistribution in any
medium, provided that the
original author and source are
credited.

INTRODUCTION

Several questions arise while browsing an unknown, yet promising journal website. How will the editorial process work? Will the submission be acknowledged as received? How long will peer review be? Will the process be beneficial for the manuscript? Will the reviews be fair? The academic research centers are overrun by terrifying publication-related experiences from colleagues. Arguably, the instant access to reviews of journals written by authors would answer those questions.

Indeed, there are proposals to write consumer reports of academic journals, e.g., (Deaner 2013; Holsinger 2013; Anderson 2013). The discussion about the tools to be employed for journal reviews is still open, and young tools like SciRev are slowly emerging. SciRev (Kochmann 2014) offers a strongly analytic, scientometrics-based platform for author-based journal reviews. The present author has contributed there but still has a preference for free-form writing. This article reviews a young, yet promising and interesting venue for publication: the Journal of Open Research Software (JORS).

THE JOURNAL OF OPEN RESEARCH SOFTWARE

The [Journal of Open Research Software](#) (JORS) is a recent proposal by Ubiquity Press. It is a *gold* open access journal as defined by Harnad et al. (2008), meaning that all the published articles are forever free to be accessed, displayed, and distributed without barriers. The journal defines itself as one, which "*features peer reviewed software papers describing research software with high reuse potential.*"

A software paper is a journal article written by following certain strict guidelines, which are [available on the journal website](#). As the guidelines state, a software paper describes an open source software with reuse potential for research purposes. The paper describes the software application, its implementation and architecture, the quality control in the development process, its availability (where to get it, how to install it), and its reuse potential within and outside the field of research of the authors.

Arguably, an immediate benefit when publishing a software paper is a gain in terms of visibility of the produced software, and the reuse by other researchers. [Figure 1](#) is JORS description of the benefits of



publishing software papers.

The two most important benefits are that *the software becomes citable* and that the citation is to a *peer reviewed paper*. In several fields, especially those falling under the general subject of computer science, writing software is an activity performed almost on a daily basis. Often, the implementation of a software system is the most time-consuming operation of research activities. With initiatives such as JORS, the open software for research becomes an active entry of the academic CV. While software is not necessarily written only by researchers in computer science, any researcher developing open source software for research would benefit from the extra journal article and the related citations. Software developers not involved in research activities, on the other hand, would arguably not care about the journal article. However, JORS is about academic publishing and research activities.

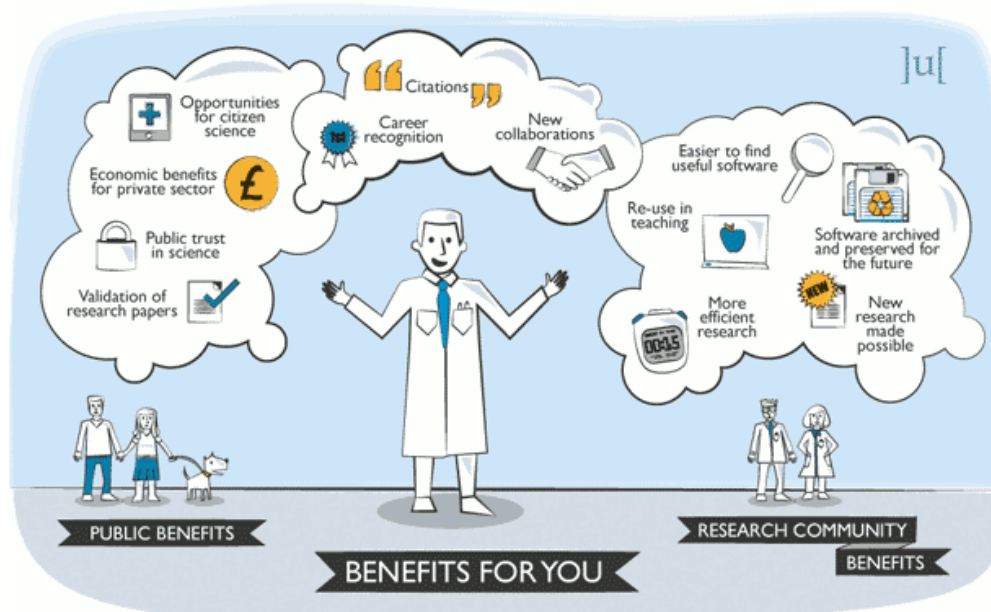


Figure 1 Journal of Open Research Software benefits for authors. Picture licensed under aCreative Commons Attribution 3.0 License. Copyright Ubiquity Press.

It should be kept in mind that a software paper is a *real* journal article. JORS is an academic journal, not a repository. *All software papers are peer-reviewed* and must satisfy the reviews' criteria and the associated editor's comments in order to be published. The review process is transparent, and the *reviewer guidelines are publicly available*. The review criteria are based on the two following principles. (1) When reviewing, the accuracy and quality of the metadata has precedence over the software, however there will be a minimum level of quality expected. (2) All papers are expected to be able to pass after revisions, unless the software is not openly available or extremely difficult to reuse. The journal does not strictly define what is the expected minimum level of quality expected from the software. Therefore, the expected quality refers to the subjective judgment of the peer reviewers.

According to the open access analysis framework described in Graziotin, Wang, and Abrahamsson (2014), the journal provides most of the desirable features of quality gold open access journals. Table 1 to Table 5 are part of the analysis framework, and describe the journal. Table 1 contains the bibliographic information of JORS. The publisher, Ubiquity Press, is established in the United Kingdom. Ubiquity Press is a relatively small publisher. However, it is well known in open access publishing communities. The publisher is part of the most famous professional affiliations, namely the Open Access Scholarly Publishers Association (OASPA), the Committee on Publication Ethics (COPE), and the Association of Learned and Professional Society Publishers (ALPSP). While Ubiquity Press is a for-profit publisher, its article processing charges are low and *transparently stated in the publisher's website*. The journal provides its articles in electronic formats. It is not printed on paper. Table 2

summarizes the activity metrics of the journal. With only 13 articles currently published (mean value of 6.50 published articles per year), it is evident that the journal is very young. It cannot be considered an established journal. **Table 3** provides the economics metrics of the journal. The article processing charges for an accepted paper are 25.00 GBP. However, the fees are partially or fully waived if the authors cannot personally afford them. As stated in **Table 4**, the authors keep the copyright of the published articles, which are released under a **Creative Commons CC-BY 3.0 license**. Each published article receives a Digital Object Identifier (DOI). As stated by Langston and Tyler (2004), a DOI is an identifier for online objects of intellectual property. The location of an online object might suddenly change. However, the DOI remains the same. Therefore, the DOI resolvers such as doi.org are able to make the DOI point to the relocated object. Many publishers include this feature, and it is becoming a necessity with electronic journals (Langston and Tyler 2004). The contents of the journal are also archived in the *LOCKSS* digital archiving system and the *CLOCKSS* distributed archive. The archival means that the articles are safe and can be retrieved even if the journal suddenly disappears. The journal lacks a disclosure of the indexing and abstracting services covering the articles [1]. Google Scholar correctly finds and indexes the journal contents. According to **Table 5**, the journal does not present any predatory factors or general issues, and it is not included in the (in)famous Beall's list of predatory journals and publishers. The start year is not an indicator of general issues because the journal is young, the article processing charges are low, and there are only 6.50 published articles per year on average.

Table 1 Bibliographic information

Name	Publisher	Country	Electronic ISSN	Printed ISSN
Journal of Open Research Software	Ubiquity Press	United Kingdom	2049-9647	No

Table 2 Activity metrics

Published articles	Average number of published articles per year	Issues per year
13	6.50	1

Table 3 Economics

Article processing charges USD	Avoidable article processing charges
42.62 (25.00 GBP)	Yes

Table 4 Accessibility

Authors keep copyright	Provides DOI to articles	Digital preservation mechanisms	Declared index	Verified index
Yes	Yes	Yes	0	1

Table 5 Predatory factors and general issues

Start year	Predatory listed	Predatory factors and issues
2013	No	No

THE PRESENT AUTHOR'S EXPERIENCE WITH JORS

Near the end of March 2013, this author was finishing the first public release of the Semat Accelerator (Graziotin and Abrahamsson 2013), which is a Web-based modeling tool for SEMAT Essence theory of Software Engineering (Jacobson et al. 2014; SEMAT 2009). The SEMAT initiative attempts to "refound software engineering based on a solid theory, proven principles, and best practices." (Jacobson et al. 2014, 46) with a kernel for describing, addressing, and measuring the common *objects* of software engineering. These objects, named alphas, are opportunity, stakeholders, requirements, software system, team, work, and way-of-working. The aim of the research project Semat Accelerator, conducted with this author's PhD supervisor Pekka Abrahamsson, was to create a prototype as awareness creator for the theory. Semat Accelerator implements Essence kernel alphas and their related states, and it represents a software engineering endeavor in a quantitative way, through the use of graphs. Additionally, the tool stores the changes in states of the alphas, and exports them as a comma-separated values file (known as CSV). The tool would ideally help practitioners and researchers to understand *Essence*, in order to evaluate its effectiveness in case studies and controlled experiments. The prototype was well received from the SEMAT community (Jacobson, Spence, and Ng 2013), especially during several conference calls with Essence core researchers and the researchers involved in expanding and using it. Therefore, it was decided to open-source the tool with the hope that research could be built on top of it and to find collaboration.

As researchers, we discussed the possibility to publish the tool in a scientific venue. Despite the research, it turned out it was difficult to publish software in our field, software engineering. The irony. Apparently, as reported by this author's supervisors and senior colleagues, the possibilities were to write a workshop paper or research paper, which employs the tool. Under fortunate conditions, people would cite those articles when employing the tool for their own research purposes.

On 2013-03-29, the tweet represented in Figure 2 was posted by the @openscience Twitter account.



Figure 2 Screenshot of a tweet sent by the @openscience announcing JORS call for papers.

The tweet opened up a call for papers at JORS. This author and his supervisor were enthusiastic to submit a software paper there.

THE SUBMISSION PROCESS

On the same day JORS was discovered, this author contacted the Editor-in-Chief, Neil Chue Hong, who kindly answered all the questions and solved all our doubts. The decision to submit a software

paper was taken almost immediately.

The software paper was written by employing the supplied templates (available as .doc and .odt for Microsoft Office and LibreOffice, respectively). Direct online writing and submission was also possible, ^ la blog post. The major difficulty we encountered was that we had no clue on what JORS would expect as the contents in the software paper. There were no published software papers in JORS website, and the templates were lacking explanations. Fortunately, the journal describes the review process and the reviewing criteria carefully, in [Figure 3](#). The review process is officially single-blinded, and the reviews are not published together with the articles. The single-blinded process was disappointing to the present author, because JORS is an innovation in the traditional academic publishing system. Innovative journals like PeerJ, Science Open Research, and The Winnower embrace open peer review at different levels. Arguably, an open access journal dealing with open source software should have an open peer review process, in order to improve its transparency [2].

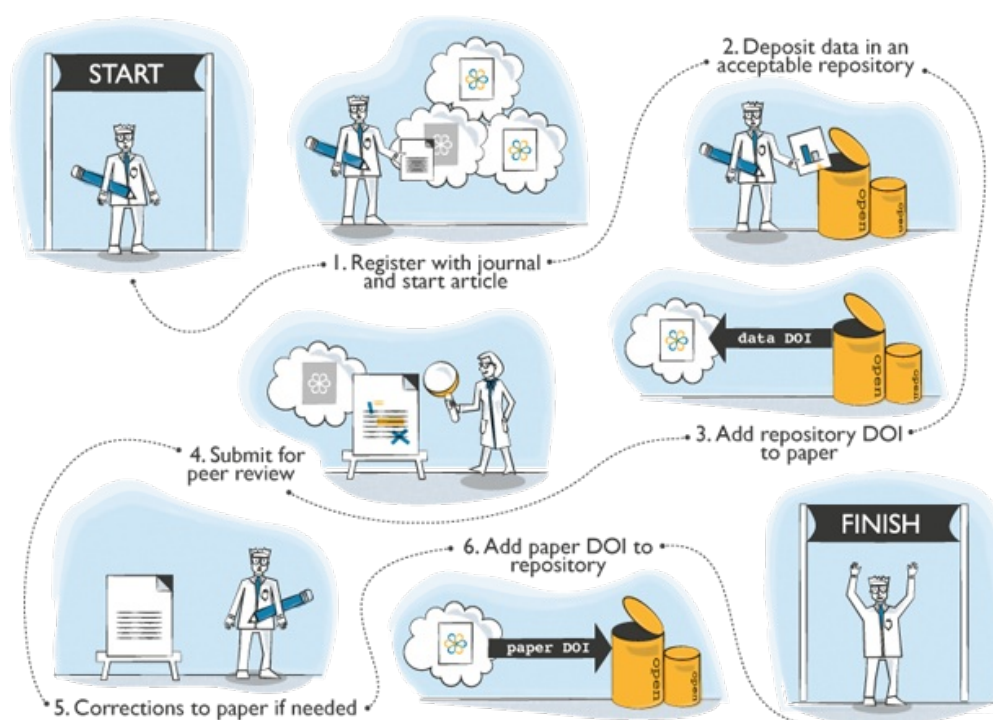


Figure 3 Journal of Open Research Software submission process. Picture licensed under a [Creative Commons Attribution 3.0 License](#). Copyright Ubiquity Press.

We submitted the manuscript on 2013-04-05 (a Friday, late afternoon). Its length was 7 pages (3 figures).

The editor acknowledged our submission on 2013-04-08 (Monday morning). We were informed that after an initial screening, the paper would be peer-reviewed. The editor's e-mail was encouraging to us, especially regarding the seriousness of the editorial team. Meanwhile, the first two JORS articles had been published. Their content was not relevant to our field, but we thought that our submission was of a fair quality in comparison.

On 2013-05-06, we contacted the editor-in-chief, asking what was typical review length. The editor promptly informed us that the manuscript was under review. They expected the reviews to arrive in "a couple of weeks." We were informed that JORS was assigned a Managing Editor, Samuel Moore from Ubiquity Press. Therefore, the editorial process would have gained benefit as there was a person thenceforth responsible for author-related enquiries and production issues. The transparency of the editor was helpful for us to perceive the difficulties when setting up a scientific journal.

We were contacted by the Managing Editor on 2013-06-18. The reviews arrived. We were positively

surprised. Both reviews' length was about 1300 lines. The length was nearly the amount the same of the one of the submitted paper (1900 lines). The feedback was harsh, because one reviewer was not a fan of SEMAT Essence—the reviewer admitted it in the review. However, it was constructive in both cases. The editorial decision was major revisions. We were provided the reviews and a brief summary of the mandatory points to be clarified with the new submission. The reviewers demanded *both a revision of the paper and the software*. This demand was unexpected and very well received by us.

On 2013-07-26, we submitted a revised version of the paper together with the updated software. The resulting work from peer-review was significantly improved. The revised article doubled in its size: 12 pages, 5 figures and 3850 words. The software remarkably changed, too, as shown by Github comparison tool (Github 2013). On the Github repository, there were 22 commits and 124 changed files, with 23,955 additions and 11,544 deletions.

On 2013-08-12, the managing editor came back to us with notification of acceptance. He offered a slightly edited version of the manuscript (with track mode on), in which he corrected minor mistakes of our non-native English. The managing editor also highlighted those sentences requiring rewording, and he provided suggestions as Microsoft Word comments.

Our article was published on 2013-09-02 (Graziotin and Abrahamsson 2013). The process summed up to 2 months, 13 days from the submission to the editorial decision. It should be noted that our submission occurred at the very beginning days of the journal. They were without a Managing Editor most of this time. Given that the second round of reviews arrived in 17 days, the review time of the journal is expected to improve.

CONCLUSIONS

The present author and his JORS co-author were pleased and satisfied with the Journal of Open Research Software. The journal has value to the scientific community. It has been a significant reward in terms of personal satisfaction to have a software system as a peer reviewed publication. Especially in computer science-related fields, the time spent on building software for research activities is significant and seldom rewarded. JORS rewards researchers who write and share their software for research activities, and permits the possibility to third party researchers to reward the software developers, with citations. The published articles are pleasant to read, with a polished layout. The people behind JORS handle a transparent process. This author felt respected and treated as a human being in a warm, constructive environment. The feedback from authors is welcome at JORS.

This author recommends submission to the Journal of Open Research Software, especially to researchers in the fields of software engineering, information systems, human-computer interaction, and the whole computer science area. Researchers build software for research continuously. Let us share it and be awarded for it.

ACKNOWLEDGEMENTS

This article is a revision of a post, which originally appeared on the author's personal website at <http://ineed.coffee/1929/review-journal-of-open-research-software>.

The author wishes to disclose that he is currently an editorial board member of the Journal of Open Research Software. However, the present review was written when such competing interest did not exist. This revision enhanced the original writing. However, the revision did not alter the structure or the meaning of the review.

The author thanks *The Winnower* for publishing this review without charges and for challenging the traditional publishing system. The author is grateful to the reviewers of this manuscript and to Elena Borgogno for the suggestions for improving this article. Finally, the author wishes to thank the people involved in [@openscience](#).

NOTES

[1] This is an issue, which is currently being addressed by the journal editorial board. JORS will be indexed in more services as soon as it reaches a fair amount of published paper. The present author learned that reaching a fair amount of papers is a common requirement for being indexed in academic databases.

[2] This is another issue under discussion by the journal editorial board. It is currently allowed to sign reviews of software papers. Indeed, the present author signed all of his reviews performed for JORS. However, this option is currently not advertised on the journal website. The possibility to publish reviews together with the software papers was well received by the editorial board, too. However, there is the need to implement it in the journal publishing system.

REFERENCES

- Anderson, Kent. 2013. "Do We Need a Consumer Reports of Journals, Written by the Authors?" *Scholarly Kitchen*. <http://perma.cc/38U2-WUUF>.
- Deaner, Robert. 2013. "It's Time for Journals to Be Author-Reviewed." *The Chronicle of Higher Education*. <http://perma.cc/69R-CB3Q>.
- Github. 2013. "Comparison of s4fs/sematacc v1.1.0 and v1.2.0." *Github*. <https://github.com/s4fs/sematacc/compare/v1.1.0...v1.2.0>.
- Graziotin, Daniel, and Pekka Abrahamsson. 2013. "A Web-Based Modeling Tool for the SEMAT Essence Theory of Software Engineering." *Journal of Open Research Software* 1 (1): e4. doi:10.5334/jors.ad.
- Graziotin, Daniel, Xiaofeng Wang, and Pekka Abrahamsson. 2014. "A Framework for Systematic Analysis of Open Access Journals and Its Application in Software Engineering and Information Systems". *Scientometrics* Online First (April 2014): 1-37. [arXiv:1308.2597](https://arxiv.org/abs/1308.2597) [cs.DL]. doi:10.1007/s11192-014-1278-7.
- Harnad, Stevan, Tim Brody, Francois Vallieres, Les Carr, Steve Hitchcock, Yves Gingras, Charles Oppenheim, Chawki Hajjem, and Eberhard R. Hilf. 2008. "The Access/Impact Problem and the Green and Gold Roads to Open Access: An Update." *Serials Review* 34 (1): 36-40. doi:10.1016/j.serrev.2007.12.005.
- Holsinger, Bruce. 2013. "A 'Consumer Reports' for Academic Journals? (and Wouldn't a Wiki Work?)." *Burnable Books*. <http://perma.cc/7UT8-TR3E>.
- Jacobson, Ivar, Pan-Wei Ng, Ian Spence, and Paul E. McMahon. 2014. "Major-League SEMAT." *Communications of the ACM* 57 (4): 44-50. doi:10.1145/2580712.
- Jacobson, Ivar, Ian Spence, and Pan-wei Ng. 2013. "Agile and SEMAT." *Communications of the ACM* 56 (11): 53-59. doi:10.1145/2524713.2524723.
- Kochmann, Sven. 2014. "SciRev - Peer-Reviewception!" *Indianalytics* 1 (1): 316. <http://www.indianalytics.de/?p=316>.
- Langston, Marc, and James Tyler. 2004. "Linking to Journal Articles in an Online Teaching Environment: The Persistent Link, DOI, and OpenURL." *The Internet and Higher Education* 7 (1): 51-58. doi:10.1016/j.iheduc.2003.11.004.
- SEMAT. 2009. "Software Engineering Method and Theory." *Semat.org*. http://semat.org/?page_id=2.