Digging into the Eclipse Marketplace

Jacob Krüger[1,2(✉)], Niklas Corr[1], Ivonne Schröter[2,3], and Thomas Leich[1,3]

[1] Harz University of Applied Sciences, Wernigerode, Germany
{jkrueger,tleich}@hs-harz.de
[2] Otto-von-Guericke University, Magdeburg, Germany
ivonne.schroeter@ovgu.de
[3] METOP Gmbh, Magdeburg, Germany

Abstract. Eclipse is an integrated development environment that can be extended with plug-ins. Thanks to Eclipse's success, a diverse community has been established with members coming from industry, open-source projects, and others, and a marketplace with more than 1.700 different plug-ins developed. Hence, the question arises how this marketplace is composed: *Who contributes plug-ins? Which plug-ins are successful? Are there common characteristics or trends?* To answer these questions, extensive investigations are necessary. In this paper, we present *(i)* an initial approach for corresponding analyses and *(ii)* preliminary results. Overall, we aim to pave the way for further research addressing, for example, motivations to participate in, or the evolution of, open marketplaces.

Keywords: Eclipse IDE · Eclipse Marketplace · Open source · Empirical study

1 Introduction

Open-source systems gained momentum in software engineering mainly because of free use, accessibility, and fast innovation speed [16,17,20,21]. As a result, many companies use such systems as basis for their own products [6], wherefore commercial and open-source software are more and more used concurrently [1,2,10,17]. A good example for this co-existence is the de facto standard *Eclipse*[1] [6,7,19].

Eclipse became the dominant integrated development environment (IDE) for Java used by developers from industry, universities, and open-source communities [5–7,9,19,23]. To support Eclipse and plug-in developers, the Eclipse Foundation implemented platforms to manage projects [5,16]. These platforms are support measures and drive the evolution of the community. As a result, the amount of plug-ins is steadily increasing from 1.385 in 2007 [23] to 1.762 at the beginning of 2017. In the context of this paper, the marketplace, in which these plug-ins are provided, will be the area of analysis.

[1] http://www.eclipse.org/, 02.01.2017.

F. Balaguer et al. (Eds.): OSS 2017, IFIP AICT 496, pp. 60–65, 2017.
DOI: 10.1007/978-3-319-57735-7_7

While the Eclipse marketplace brings together diverse communities it has rarely been analyzed. However, such an analysis can help researchers to understand what motivates developers with different backgrounds (e.g., open-source and industry) to participate and collaborate. Furthermore, the findings can support to scope plug-ins and initiate cooperation. In this paper, we describe a preliminary analysis of the Eclipse marketplace and corresponding results. Overall, we aim to provide a glance on open marketplaces and initiate more detailed research.

2 Research Method

Different approaches can be used to analyze the Eclipse marketplace, for instance, empirical studies based on interviews or questionnaires. However, to scope further research, we propose to mine and assess data available in the marketplace, providing a starting point based on the users' perspective. To this point, we address two research questions within this article:

RQ-1 Which topics are addressed by successful plug-ins? Plug-ins and topics that accumulate more attention (i.e., more downloads) indicate practical acceptance. This analysis can help to select suitable plug-ins or to scope further development, for instance to improve existing approaches.

RQ-2 Who contributes to these plug-ins? Based on the previous question, we investigate who develops these plug-ins. This can help to identify leading developers and communities for specific topics, potentially indicating collaborations and new research directions.

We automatically crawled the marketplace and manually analyzed the data. For this, we limited our analysis to the 100 most downloaded plug-ins until October 2016, covering approximately 80.3% of all downloads until then. We remark, that our methodology in this article is preliminary and shall only provide a starting point for further research.

3 Preliminary Results

In Fig. 1, we illustrate the topics we identified within our sample and their categorized owners. Furthermore, we show the distribution of downloads for each topic, and especially for open-source communities.

RQ-1: Which topics are addressed by successful plug-ins? Developers often use several and synonymous terms to describe their plug-ins, hampering an automated categorization. Thus, we manually assessed the purpose of the 100 most downloaded plug-ins and derived 10 initial categories: `Revision control`, `IDE extension` (integrating programming languages or frameworks), `code analysis`, `build tool`, `user interface`, `database`, `editor`, `optimization` (of Eclipse), `documentation`, and `server`. As we show in Fig. 1, revision control and

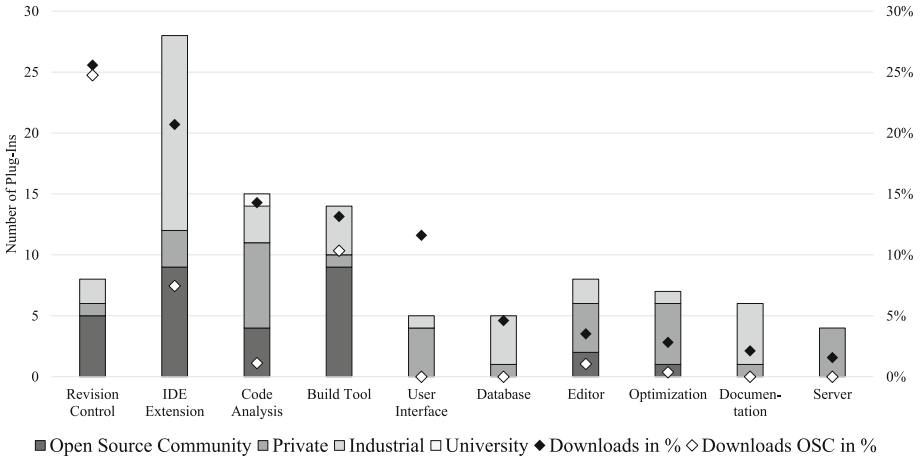

Fig. 1. Categorized projects of the 100 most downloaded plug-ins in the Eclipse marketplace.

IDE extensions accumulate most downloads. However, we also see that there is a difference in the number of available plug-ins for these two categories.

Only 8 plug-ins address revision control but are responsible for over a quarter of all downloads. This might be explained with a small set of established approaches, similar to the situation for user interfaces. Still, this result can be distorted, for instance because communities may require identical tooling, forcing participants to use a specific system even if they prefer another one.

In contrast, IDE extensions, similar to several remaining categories (e.g., code analysis, or build tools), provide far more plug-ins (28) while accumulating fewer downloads. A potential explanation is that these plug-ins often require adaptations to specific programming languages and development processes. Hence, their overall applicability is limited to specific communities and they compete with other IDEs.

RQ-2: Who contributes to these plug-ins? To gain an impression of Eclipse's community, we categorized the owners of plug-ins into four groups: `open-source`, `private`, `industrial`, and `university`. As we see in Fig. 1, open-source communities are leading in revision control and build tools, accumulating most plug-ins and downloads. Especially in revision control, which is used for distributed work and collaboration, they seem to benefit from their diversity [5]. However, open-source communities do not, or rarely, contribute to some other topics in our sample.

In contrast, private and industrial owners provide plug-ins for almost all topics. Private developers seem to dominate the development of user interfaces, optimization, and server integration, which facilitate using Eclipse. However, such plug-ins provide less utility besides comfort than, for example, customized build tools or revision control. For this reason, industrial and open-source

communities might be less interested in developing these. We support this argument due to the fact that industrial owners provide most plug-ins in IDE extensions, databases, and documentation. These are essential aspects of software development in companies. Finally, we remark that universities own only a single plug-in in our sample, potential reasons being that they provide innovative but immature projects.

4 Research Agenda

The results presented in this work are preliminary and further analysis on how such marketplaces are composed are necessary. More detailed investigations may help to understand, why communities do or do not provide plug-ins and what motivates them. We emphasize that this requires extensive analyses and additional empirical studies. In future work, we aim to assess the following aspects:

How to assess the success of a plug-in? In this paper, we solely focused on the number of overall downloads. This is a significant limitation and other metrics are necessary to provide a more detailed view. For instance, to consider the marketplace's evolution, a plug-in's downloads in recent periods, integration into Eclipse packages, or the number of its developers are interesting.

Are there common characteristics of successful plug-ins? Different communities successfully participate in the Eclipse marketplace. Hence, success may depend on the topic and also certain characteristics, for example, used licenses [13,17,18], necessity to pay, or maturity. For community managers and developers such information are important to design and provide their software.

How do users select plug-ins? The previous two questions may indicate, and can be validated by investigating how, users select plug-ins. Besides these points, the users' selection also depends on their experiences and background. The results can help to understand how new techniques emerge and establish.

Why and how are plug-ins developed? An important question in the context of open-source software is the motivation of developers. In particular, it is interesting who initiates plug-ins for which reason, who contributes to these, or which connections exist. The corresponding results may provide insights into motivations to collaborate and participate in communities and open marketplaces.

5 Related Work

Several authors investigate the evolution of Eclipse or its plug-ins, focusing, for instance, on their architecture or API usage [3,4,14,22]. Further works investigate the laws of software evolution [12] in open-source systems [8,11]. In contrast to the systems themselves, our scope is how a potential marketplace for these evolves and is composed. Still, both approaches are complementary and can be combined, for instance to assess how a plug-ins evolution affects its popularity and status. Finally, Murphy et al. [15] empirically evaluate how developers use the Eclipse IDE. It seems interesting to utilize this approach to also assess how plug-ins are used.

6 Conclusions

Eclipse is one of the most prominent and widely used Java IDEs. Due to its success, a large and diverse community of plug-in developers established. They provide their plug-ins at the Eclipse marketplace, allowing other users to use them.

In this paper, we proposed to analyze the Eclipse marketplace. The preliminary results show that some topics, such as revision control or IDE extensions, are often demanded by users and that different owners participate. Finally, we described further research directions to deepen the understanding of open marketplaces and their communities.

Acknowledgments. This research is supported by DFG grant LE 3382/2-1 and Volkswagen Financial Services AG. We thank Heike Fischbach for commenting on drafts of this paper.

References

1. Bonaccorsi, A., Rossi, C.: Why open source software can succeed. Res. Policy **32**(7), 1243–1258 (2003). doi:10.1016/S0048-7333(03)00051-9
2. Bonaccorsi, A., Rossi, C.: Comparing motivations of individual programmers and firms to take part in the open source movement: from community to business. Know. Technol. Policy **18**(4), 40–64 (2006). doi:10.1007/s12130-006-1003-9
3. Businge, J.: Co-evolution of the Eclipse SDK framework and its third-party plug-ins. In: European Conference on Software Maintenance and Reengineering, CSMR, pp. 427–430. IEEE (2013). doi:10.1109/CSMR.2013.64
4. Businge, J., Serebrenik, A., van den Brand, M.: An empirical study of the evolution of Eclipse third-party plug-ins. In: Joint Workshop on Software Evolution and International Workshop on Principles of Software Evolution, IWPSE-EVOL, pp. 63–72. ACM (2010).doi:10.1145/1862372.1862389
5. des Rivières, J., Wiegand, J.: Eclipse: a platform for integrating development tools. IBM Syst. J. **43**(2), 371–383 (2004). doi:10.1147/sj.432.0371
6. Ebert, C.: Open source software in industry. IEEE Softw. **25**(3), 52–53 (2008). doi:10.1109/MS.2008.67
7. Geer, D.: Eclipse becomes the dominant java IDE. Computer **38**(7), 16–18 (2005). doi:10.1109/MC.2005.228
8. Godfrey, M.W., Tu, Q.: Evolution in open source software: a case study. In: International Conference on Software Maintenance, ICSM, pp. 131–142. IEEE (2000). doi:10.1109/ICSM.2000.883030
9. Goth, G.: Beware the March of this IDE: Eclipse is overshadowing other tool technologies. IEEE Softw. **22**(4), 108–111 (2005). doi:10.1109/MS.2005.96
10. Hars, A., Ou, S.: Working for free? Motivations for participating in open-source projects. In: Hawaii International Conference on System Sciences, HICCS, pp. 1–9. IEEE (2001). doi:10.1109/hicss.2001.927045
11. Koch, S.: Software evolution in open source projects - a large-scale investigation. J. Softw. Maint. Evol. **19**(6), 361–382 (2007). doi:10.1002/smr.348
12. Lehman, M.M., Ramil, J.F.: Rules and tools for software evolution planning and management. Ann. Softw. Eng. **11**(1), 15–44 (2001). doi:10.1023/A:1012535017876

13. Manabe, Y., German, D.M., Inoue, K.: Analyzing the relationship between the license of packages and their files in free and open source software. In: Corral, L., Sillitti, A., Succi, G., Vlasenko, J., Wasserman, A.I. (eds.) OSS 2014. IAICT, vol. 427, pp. 51–60. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55128-4_6
14. Mens, T., Fernández-Ramil, J., Degrandsart, S.: The evolution of Eclipse. In: International Conference on Software Maintenance, ICSM, pp. 386–395. IEEE (2008). doi:10.1109/ICSM.2008.4658087
15. Murphy, G.C., Kersten, M., Findlater, L.: How are java software developers using the elipse IDE? IEEE Softw. **23**(4), 76–83 (2006). doi:10.1109/MS.2006.105
16. Parreiras, F.S., Gröner, G., Schwabe, D., de Freitas Silva, F.: Towards a marketplace of open source software data. In: Hawaii International Conference on System Sciences, HICSS, pp. 3651–3660. IEEE (2015). doi:10.1109/HICSS.2015.439
17. Ramanathan, L., Iyer, S.K.: A qualitative study on the adoption of open source software in information technology outsourcing organizations. In: Damiani, E., Frati, F., Riehle, D., Wasserman, A.I. (eds.) OSS 2015. IAICT, vol. 451, pp. 103–113. Springer, Cham (2015). doi:10.1007/978-3-319-17837-0_10
18. Ruffin, M., Ebert, C.: Using open source software in product development: a primer. IEEE Softw. **21**(1), 82–86 (2004). doi:10.1109/MS.2004.1259227
19. Vaughan-Nichols, S.J.: The battle over the universal java IDE. Computer **36**(4), 21–23 (2003). doi:10.1109/MC.2003.1193223
20. von Hippel, E.: Innovation by user communities: learning from open-source software. Sloan Manage. Rev. **42**(4), 82–86 (2001)
21. Watson, R.T., Boudreau, M.-C., York, P.T., Greiner, M.E., Wynn Jr., D.: The business of open source. Commun. ACM **51**(4), 41–46 (2008). doi:10.1145/1330311.1330321
22. Wermelinger, M., Yu, Y.: Analyzing the evolution of Eclipse plugins. In: International Working Conference on Mining Software Repositories, MSR, pp. 133–136. ACM (2008). doi:10.1145/1370750.1370783
23. Yang, Z., Jiang, M.: Using Eclipse as a tool-integration platform for software development. IEEE Softw. **24**(2), 87–89 (2007). doi:10.1109/MS.2007.58