Attracting, Onboarding, and Retaining Newcomer Developers in Open Source Software Projects

Igor Steinmacher
Department of Computing
Federal University of Technology – Paraná (UTFPR)
Campo Mourão – PR – Brazil
igorfs@utfpr.edu.br

Marco Aurélio Gerosa
Department of Computer Sciences
University of São Paulo (USP)
São Paulo – SP – Brazil
gerosa@ime.usp.br

David F. Redmiles
Department of Informatics
University of California, Irvine (UCI)
Irvine – CA - USA
redmiles@ics.uci.edu

ABSTRACT

Open source software (OSS) is a good example of global software development. Numerous OSS projects depend on contributions from volunteers from all over the world. For OSS projects, keeping a good influx of new developers is critical. In this paper, we present a *developer joining model* that represents the stages that are common to and the forces that are influential to newcomers being drawn or pushed away from a project. We claim that, studying these aspects, especially the hindering factors, can lead to new challenges and research opportunities around awareness, coordination practices, socialization, knowledge management, and possibly others.

Author Keywords
Newcomers; Open Source Software; Joining Process

ACM Classification Keywords
• **Human-centered computing ~ Open source software**
• *Software and its engineering ~ Collaboration in software development*

INTRODUCTION

Volunteers from different parts of the globe support various Open Source Software (OSS) communities. According to Qureshi and Fang [14], it is essential to motivate, engage, and retain new developers in order to promote a sustainable number of developers in a project. Therefore, a major challenge for OSS projects is to provide ways to support the joining of newcomers.

The term *joining* appears in the literature [3, 6, 8] as a way to define the process of a developer becoming a core member of a project, or, the transition from being an outsider to being an insider, i.e., a member of the project. A common way to model OSS communities and the joining process is the so-called onion model. This model is presented as a set of concentric layers (like the layers of an onion) representing how actors are positioned in communities as core developers, active developers, bug fixers, bug reporters, mailing list contributors, or plain users.

In the onion model of the joining process, newcomers join a project by first contributing through mailing list discussions and bug trackers and as they develop skills and reputation within the community, they advance to central roles for contributing code and making design decisions.

The onion model is a well-accepted model to map the social structure and the project roles in an OSS project. However, using it to map the steps of a joining process has been questioned in some studies [6, 12]. Herraiz et al. [6], for example, analyzed the contribution history of the GNOME project and found that "…most of the developers have committed a change before they ever sent a bug report (52.5%)… In general, the entire population of developers does not comply with the onion model…bug reports appear before messages in the mailing list."

We claim that joining a project is a complex process composed of different stages and a set of forces that push newcomers towards or away from the project. We deliberately break the joining process into two stages: onboarding and contributing, since there are different emphases in each one of them. While onboarding stage is highly impacted by a steep learning curve as well as reception and expectation breakdowns, it is longer-term forces influence the contributing stage. Moreover, not every developer wants to become a contributor, committer, or a core member although everyone faces the problems of onboarding before making their first contribution. We also highlight the importance of what comes before the joining process. We characterize the forces that draw outsiders to a project, such as motivation and project attractiveness. While motivation persists as an ongoing force, various hindering factors and retention forces influence onboarding, contribution, and members' permanence. In this paper, we represent these forces and stages in a preliminary model and position the state-of-the-art within it.

DEVELOPERS JOINING MODEL

We propose a model representing the stages and forces that influence the joining process of a developer to an OSS project. This model is presented in Figure 1. This model is composed of the stages that are common to and the forces that are influential to the joining process. We will briefly explain each of these elements in this section.

Before presenting the model in detail, it is worth to notice that we are aware that there are different ways to start contributing to an OSS project. This includes translation, bug triaging, bug reporting, user support, and source code contributions. In this paper, we are focusing only in source code contributions, which is a piece of code – that implements a new feature or fixes some bug – successfully sent and properly accepted by the project.

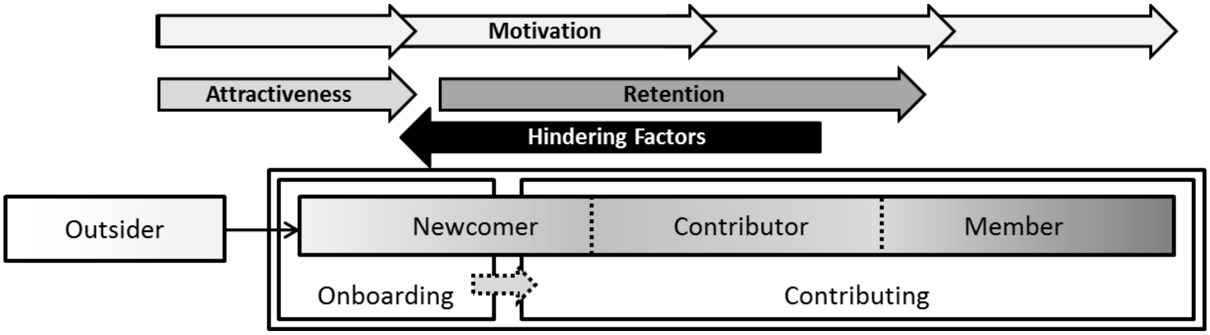

Figure 1. Developers joining model presenting the stages and forces that act during the joining process

The central elements of the model are the stages that developers go though and for which OSS communities should have different investments in terms of having more developers contributing to the project. An *outsider* represents a potential contributor to the project who is not involved with the development so far. A *newcomer* is a developer trying to place their first code contributions into the project. A *contributor* represents a developer that participates in the project, but who is not recognized as a member and does not have commit privileges. A *member* is someone recognized by the community as a developer or formal contributor.

We represent in the model four different forces that influence the progress from one stage to the following. Motivation and project attractiveness are the triggers to push the outsider to contribute to the project. *Motivation* forces represent internal (e.g., learning, self-marketing, recognition) and external (e.g., scholarship, course assignment, feature need) motives that drive a developer to join (and keep contributing) to a project. Hence, motivation forces are presented in the whole joining model, since lack of motivation lead to drop offs. The motivation forces can change/evolve during the development process. For example, some developers onboard a project because of a short-term scholarship, such the one given by Google Summer of Code, or a grade in a university course, and after that, they remain contributing to learn and self-promote themselves.

Attractiveness forces represent the characteristics and actions that the project presents to bring new users and developers [15]. These forces can include type of license, project visibility, project age, number of developers, etc.

Attractiveness and motivation work together to push outsiders toward the projects. In some cases, attractiveness forces play a special role, pulling motivated developers that did not decide which project to support.

The transition from outsider to newcomer occurs when a developer decides to contribute to a project. At this point, the developer starts onboarding the project. During onboarding, motivation keeps pushing the developer towards the project. However, some opposite forces, which we call hindering factors, can hamper developers joining process. These forces comprise technical and non-technical factors, including learning curve, lack of support from the community, difficulties finding how to start, etc.

Understanding how to deal with these hindering factors is critical to the joining process. These forces can be powerful enough to lead developers to give up contributing to the project [18]. An important thing to observe is that these factors influence both developers willing to make a single contribution and those willing to climb higher and become a member of the project.

On the other hand, retention forces may help to push newcomers to stay willing to contribute. *Retention* forces represent the characteristics and actions that a project presents to bring/keep more contributors to the project. Some of these forces are initiatives to support newcomers overcoming the hindering factors (such as providing tools to facilitate code understanding, or indicating good tasks or pieces of code to start with). Other forces represent mechanisms to support existent contributors to contribute more, triggering, in some cases, motivation change (e.g., granting commit rights to a developer, using *gamification* elements).

STATE-OF-THE-ART

In this section, we discuss the state-of-the-art of developers joining process and from the perspective of the forces represented in the proposed model.

Joining process

Besides the Onion Model, other studies tried to map the joining process on OSS projects. Von Krogh et al. [8], for example, proposed the concept of "joining script." They create this concept analyzing interviews with developers, analysis of emails, source code repository, and documents from data from project FreeNet. The proposed joining script is defined as the level and type of activity a newcomer goes through to become a member of a community.

In the same sense, Ducheneaut [3] analyzed the mailing list archives of Python project. He made an in-depth analysis of the socialization history of one successful newcomer in this community. Based on this individual, the author identified a set of socialization activities that contributed to his success in the project. They highlight the political size of the project and the existence of specific rites of passage during his trajectory.

Existing literature usually present the joining process as a set of steps and activities newcomers must perform to become a member of the project. Moreover, these studies focus on "how to become a core member." We argue that, due to the voluntary and collaborative nature of OSS projects, defining a model composed of steps or activities cannot be the best way to map developers joining process. Our approach differs from the previous ones as we focus on the forces that influence the developers' onboarding and contribution.

Motivation

Motivation represents the forces that drive developers to place their contributions to the project. They include the motives to place the first contributions and to remain contributing to the project.

Lakhani and Wolf [9] studied the motivations of individuals to contribute to OSS projects. They surveyed 684 OSS developers and found that external motivational factors in the form of extrinsic benefits (e.g.; better jobs, career advancement) are the main motivations. They also report that enjoyment, challenges derived from writing code, and improving programming skills are top motivators for contributing.

Hars and Ou [5] also used a survey to understand what internal and external motives drive OSS developers to participate in OSS projects. They analyzed data obtained from 389 developers and found that internal motivation, such as intrinsic motivation, having fun, and identification with a community play a role, but external factors have greater weight. They highlight building human capital and personal needs for a software solution as some of the main external motivations.

Shah [17] analyzed interviews, mailing list archives and documentation of two OSS projects to understand the motivation of developers. They classified the contributors in two types: need-driven and hobbyists. The motives of need-driven participants varies, and comprises necessity of a specific feature, reciprocity to the project, desire to integrate their own source code, career concerns, and getting feedback and improvements to their solutions. Hobbyists contribute to have fun and entertainment.

Attractiveness

Attractiveness represents forces that projects put in order to foster contributions from new developers.

Santos et al. [15] defined a theoretical cause-effect model for attractiveness as a crucial construct for OSS projects, proposing their typical causes (license type, intended audience, type of project, development status); indicators (hits, downloads, members); and consequences (# of open tasks, time for task completion). They tested the model with data from more than 4000 projects and found that project conditions and characteristics–such as license restrictiveness and their available resources–directly influences the attractiveness.

Meirelles et al. [10] built upon Santos' model inserting source code as one of the typical causes. They observed the influence of structural complexity and software size (lines of code and number of modules) on project attractiveness. Their results indicated that structural complexity has a negative influence and software size has a positive influence on attractiveness.

Chegalur-Smith et al. [1] analyzed whether codebase size, project age and niche size (a measure borrowed from ecology) influences project attractiveness. They found that these three characteristics influence the project's ability to attract and retain developers.

Ververs et al. [19] mapped the influential factors that determine developer participation on the Debian project. They analyzed potentially influential events and the commits of 11 years of the project. They found that the highest influences were specific events, such as CeBIT, Debian Day, new or frozen releases, incidents, dependency issues and the introduction of new developer services.

Hindering factors

Hindering factors are the forces that pose obstacles to newcomers and contributors willing to place their contributions to the project. These forces can delay their contribution and even make them give up contributing.

Jensen et al. [7] analyzed mailing lists of OSS projects to verify if the emails sent by newcomers were quickly answered, if gender and nationality influenced the kind of answer received, and if the reception of newcomers was different in users and developers lists. They found that receiving timely responses was positively correlated with future participation. Moreover, they found few rude replies to newcomers, but they report that this flaming behavior can have a chilling effect, since the mailing list is public.

Steinmacher et al. [18] used data from mailing list and issue tracker to study how reception influences the retention of newcomers in an OSS project. Among the factors that influence the decision to abandon, they found evidence that receiving inadequate answers and the experience of the respondent affect the decision of newcomers. A questionnaire sent to the dropouts revealed newcomers unhappy with the answers received, because they could not find the support needed to start.

Midha et al. [11] studied the impact of code cognitive complexity, in terms of cyclomatic complexity, in the number of new developers that contributed to the project. They gathered data from source code history of 450 projects in Source Forge. They found that an increase in cognitive complexity decreases the contributions from new developers.

Retention

Retention forces represent the ability and characteristics that a project presents to support newcomers onboarding and to keep developers contributing to the project.

Schilling et al. [16] used the concepts of Person-Job fit and Person-Team fit to evaluate the retention of former Google Summer of Code students in KDE project. They found that the development experience and students' familiarity with the project's coordination practices are strongly associated with retention. They also report that students with abilities that are underused in the project and students with a higher academic education do not remain considerably longer.

Zhou and Mockus [21] worked on identifying the newcomers who are more likely to remain contributing to the project in order to offer active support for them to become long-term contributors. They found that the interaction between individual's attitude and project's climate are associated with the odds that an individual would become a valuable contributor.

Fang and Neufeld [4] built upon Legitimate Peripheral Participation theory to understand the motivation of developers to remain contributing in a sustainable way. Results from qualitative analyses revealed that initial conditions to participate did not effectively predict long-term participation, but that situated learning and identity construction behaviors were positively linked to sustained participation.

Qureshi and Fang [14] analyzed the trajectories of 133 newcomers in 40 projects since the moment they onboard. They identified four distinct classes of newcomer behavior, considering their initial amount of interactions with core members and the growth of these interactions.

There are also some studies presenting tools to support newcomers' first steps. Čubranić et al. [2] presented Hipikat, a tool that supports newcomers by building a group memory and recommending source code, emails messages, and bug reports to support newcomers. Wang and Sarma [20] present a Tesseract extension to enable newcomers to identify bugs of interest, resources related to that bug, and visually explore the appropriate socio-technical dependencies for a bug in an interactive manner. Park and Jensen [13] show that visualization tools support the first steps of newcomers in an OSS project, helping them to find information more quickly.

CONCLUSION

In this paper, we presented a *developer joining model* aimed at representing the stages that are common to and the forces that are influential to newcomers being drawn or pushed away from an OSS project. The goal of this model is to present a new perspective on the joining process, different from those that focus on project roles or the ways of becoming a core member. As with every model, ours is a simplification or abstraction of reality. Nevertheless, we claim that it is useful for OSS communities to plan investments in better supporting new members joining.

By presenting the opposing forces in the proposed model, we aim at fostering research on weakening the hindering factors. An in-depth study of the hindering factors can bring to light new challenges, and provide insights to further research. To understand these factors we are conducting a series of interviews with core members of and newcomers to OSS projects.

Future challenges can include improving newcomers' awareness on project related information, properly managing project knowledge to support newcomers, building recommender tools, and studying social interactions during onboarding.

ACKNOWLEDGMENTS

The authors thank Fundação Araucária for the financial support. Marco Gerosa receives a grant from the CNPq and FAPESP. Igor Steinmacher receives grant from CAPES (BEX 2038-13-7). David Redmiles receives a grant from the US National Science Foundation (#1111446).

REFERENCES

1. Chengalur-Smith, I.N., Sidorova, A. and Daniel, S.L. 2010. Sustainability of Free/Libre Open Source Projects: A Longitudinal Study. J. AIS. 11, (2010).
2. Cubranic, D., Murphy, G.C., Singer, J. and Booth, K.S. 2005. Hipikat: a project memory for software development. IEEE Transactions on Software Engineering. 31, (2005), 446–465.
3. Ducheneaut, N. 2005. Socialization in an Open Source Software Community: A Socio-Technical Analysis. Computer Supported Cooperative Work (CSCW). 14, (2005), 323–368.
4. Fang, Y. and Neufeld, D. 2009. Understanding Sustained Participation in Open Source Software Projects. J. Manage. Inf. Syst. 25, (Apr. 2009), 9–50.
5. Hars, A. and Ou, S. 2001. Working for free? Motivations of participating in open source projects. 34th Annual Hawaii Intl. Conference on System Sciences (2001), 9 pp.
6. Herraiz, I., Robles, G., Amor, J.J., Romera, T., Barahona, J.M.G. and Carlos, J. 2006. The processes of joining in global distributed software projects. 2006 Intl. Workshop on Global Software Development for the Practitioners (2006), 27–33.
7. Jensen, C., King, S. and Kuechler, V. 2011. Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. 44th Hawaii Intl. Conf. on System Sciences (2011), 1–10.
8. Krogh, G. von, Spaeth, S. and Lakhani, K.R. 2003. Community, joining, and specialization in open source software innovation: a case study. Research Policy. 32, (2003), 1217–1241.
9. Lakhani, K.R. and Wolf, R.G. 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. Perspectives on Free and Open Source Software. MIT Press.
10. Meirelles, P., Santos, C., Miranda, J., Kon, F., Terceiro, A. and Chavez, C. 2010. A study of the relationships between source code metrics and attractiveness in free software projects. Software Engineering (SBES), 2010 Brazilian Symposium on (2010), 11–20.
11. Midha, V., Palvia, P., Singh, R. and Kshetri, N. 2010. Improving open source software maintenance. Journal of Computer Information Systems. 50, (2010), 81–90.
12. Oezbek, C., Prechelt, L. and Thiel, F. 2010. The Onion Has Cancer: Some Social Network Analysis Visualizations of Open Source Project Communication. 3rd Intl. Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (2010), 5–10.
13. Park, Y. and Jensen, C. 2009. Beyond pretty pictures: Examining the benefits of code visualization for Open Source newcomers. 5th Intl. Workshop on Visualizing Software for Understanding and Analysis (2009), 3–10.
14. Qureshi, I. and Fang, Y. 2011. Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. Org. Res. Methods. 14, (2011), 208–238.
15. Santos, C., Kuk, G., Kon, F. and Pearson, J. 2013. The Attraction of Contributors in Free and Open Source Software Projects. J. Strateg. Inf. Syst. 22, (Mar. 2013), 26–45.
16. Schilling, A., Laumer, S. and Weitzel, T. 2012. Who Will Remain? An Evaluation of Actual Person-Job and Person-Team Fit to Predict Developer Retention in FLOSS Projects. 45th Hawaii Intl. Conf. on System Science (HICSS) (2012), 3446–3455.
17. Shah, S.K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. Manage. Sci. 52, (2006), 1000–1014.
18. Steinmacher, I., Wiese, I., Chaves, A.P. and Gerosa, M.A. 2013. Why do newcomers abandon open source software projects? 6th Intl. Workshop on Coop. and Human Aspects of Software Engineering (CHASE) (2013), 25–32.
19. Ververs, E., Bommel, R. van and Jansen, S. 2011. Influences on Developer Participation in the Debian Software Ecosystem. Intl. Conference on Management of Emergent Digital EcoSystems (San Francisco, California, 2011), 89–93.
20. Wang, J. and Sarma, A. 2011. Which Bug Should I Fix: Helping New Developers Onboard a New Project. 4th Intl. Workshop on Cooperative and Human Aspects of Software Engineering (2011), 76–79.
21. Zhou, M. and Mockus, A. 2012. What Make Long Term Contributors: Willingness and Opportunity in OSS Community. 2012 Intl. Conf. on Software Engineering (2012), 518–528.