GUEST EDITORS: Tom Mens, Daniela Damian, Marcelo Cataldo

The Social Developer: The Future of Software Development

What characterizes an influencer in software ecosystems?

Victor Farias, *Federal University of the State of Rio de Janeiro*

Igor Wiese, *Federal University of Technology - Paraná*

Rodrigo Santos, *Federal University of the State of Rio de Janeiro*

ABSTRACT. Software development has become more social. Influence emerges as an important factor since developers use to interact over a technological platform, creating software ecosystems. Influence is the power that something or someone can have over some other thing or people. In this work, we present a survey on the developers' sense of influence in ecosystems based on GitHub repositories aiming to identify and evaluate influencers' characteristics. 3,419 developers involved in the npm package management ecosystem were invited to answer our survey. We received 242 answers in 30 days. Results show that active participation and long-time interaction with a project are crucial for characterizing an influencer. We realized that collaborators and integrators have similar perceptions about influence in ecosystems.

KEYWORDS. Software Ecosystems, Influence, Social Developer.

Influence in Software Ecosystems

Software development tends to be unequal when analyzed from a contributor perspective. A critical factor is the rising of influencers who begin to lead development and dictate how the project evolves. By studying influencers, we can understand the evolution of a software ecosystem (SECO) and predict the future of its platform. Taking GitHub projects as our object of study, a SECO refers to "groups of projects that are developed and co-evolve in the same ecosystem"[1], e.g. npm SECO is formed by GitHub projects that are related to its libraries and packages[2].

Influence, as a noun, is "the power to have an effect on people or things, or a person or thing that is able to do this"[3]. As a verb, influence is defined as "to affect or change how someone or something develops, behaves, or thinks"[3]. Influence is a kind of power: to have power over someone is to get someone to do something that he/she would not otherwise do[4]. Based on this concept, we investigated how to characterize actors that have an effect (social/technical) on people or things (artifacts), whose actions affect how the ecosystem works. Understanding actors that influence others is relevant for project owners and SECO managers[5]. These actors are responsible for monitoring the ecosystem health, the open source software value, and the confidence the community has in the platform towards attracting developers and users. A bad influence could lead a SECO to the ground just as a good influence could make a project a big success[6,7].

As reported in previous studies, focusing on a specific SECO subset/type and studying different aspects in depth is important for bringing more realistic results[4,8]. Despite the set of ecosystems studied, few works investigated GitHub repositories in such a perspective[2,8]. Therefore, this study aims to identify which characteristics a community of developers recognizes as the most relevant for any influencer. When the identification of influencers gets easier, we can help contributors to understand how an ecosystem evolves based on influencers' actions. Different from the existing literature, we investigate the sense of influence from two SECO actors' roles and from multiple characteristics.

What Characterizes an Influencer?

We performed a systematic mapping study[5] to identify characteristics of an influencer in SECO following Kitchenham et al.'s guidelides[9]. We identified 214 papers as a result of searching on influence and software ecosystem (and synonyms) in four digital libraries (IEEE-Xplorer, ScienceDirect, ACM-DL, and SpringerLink), and by reading abstracts. After a second revision, when we read introductions/conclusions, 15 papers were selected: nine presented characteristics of an influencer in SECO (some from GitHub), and six were related work (mappings). From those papers, we extracted eight relevant characteristics related to the social interaction, or to technical code contribution. Influencers' characteristics are described in Table 1.

TABLE 1. Influencers' Characteristics

Characteristic	Description
Closeness to the GitHub Project Owner	The closer a developer is to a GitHub project owner, the easier he/she influences the ecosystem, as his/her contribution and changes would be more likely to be accepted.
Long-time Interaction with the Project	As a developer interacts with a project for a long time, such actions increase his/her chance to influence a SECO by comments or code contribution.
Status in the Project	It would be easier for a developer to influence a SECO as an integrator on a GitHub project as long as this role is responsible for accepting contributions and having a shorter way to making the contribution integrated to the project.
Status (popularity on GitHub)	The more followers a developer has, the easier he/she influences a SECO as his/her status could lead other developers to follow his/her vision, and his/her contribution would be more likely accepted.
Content Value	Recognition as a source of high value contribution would help a developer to have his/her contributions accepted as well as his/her influence spread.
Source of Learning	Recognition as someone whose knowledge in some technical aspect would be enough for others to adopt techniques/support ideas of his/her reference.
Participation with Code	Participation with code contribution would influence directly the ecosystem, as it could lead the project to different paths related to the influencer' interests.
Participation with Comments	Participation with comments could lead SECO developers to contribute to the influencer's interests, e.g. demand for a new functionality.

Survey Method

Our objective was to gather information on how developers realize influence in the ecosystem they participate. We chose the npm SECO as it was characterized as an ecosystem in previous research[2] and has a huge number of projects/interdependencies. Another reason is the possibility to collect socio-technical data available at GitHub.

Free/Libre Open Source Software (FLOSS) development team structure was already discussed by Crowston et al.[7]. From their perspective, integrators (committers) are core developers (who contribute frequently and manage project evolution) and collaborators are co-developers (who contribute sporadically with code review and bug fixes). As we are investigating influence in SECO, both actors somehow act as influencers in a project. We consider developers who contribute to npm projects with at least one merged pull request as *collaborators*. By *integrators*, we refer to developers with permission to merge pull request in the projects. Based on these roles, we prepared a survey to analyze the sense of influence in npm SECO. This survey was designed based on Kitchenham et al.'s recommendations[9], such as sending personalized invitations, allowing participants to remain anonymous, and asking direct questions. Open questions were analyzed in pairs, followed by conflict resolution meetings.

Developers were divided in two groups (collaborators/integrators) so that differences could be identified[7]. We created an infrastructure to download metadata (name, author, release date, GitHub link, number of downloads) for all packages available in npm[10]. Next, we collected email addresses through GitHub API. In September 2017, npm SECO had over 510,964 packages (https://github.com/utfpr/EcosystemsAnalysis) from which we randomly extracted 663 projects (confidence level of 99% with a margin of error of 5%). We removed 46 projects from our sample because they did not provide a GitHub link at npm. Thus, we randomly selected another sample of 46 projects.

Our survey had four demographic questions (DQs) and three technical questions (TQs). DQs were: (1) *what is your gender? (Male, Female, Gender Variant/Non-Conforming, Prefer not to answer)*; (2) *Are you a… (Student, Professional, Postdoc, Professor)*; (3) *Do you have a Computer Science Background? (Yes, other…)*; and (4) *How many projects did/do you contribute to? (1, 2, 3, 4, 5+)*. TQs were: (A) *In your point of view, what makes anyone an influencer in a GitHub repository?*; (B) *What is your influence level in the projects you are involved with? Justify your previous answer.*; and (C) *What users do you see as influencers of your GitHub projects?* Regarding TQ-A, options were influencers' characteristics from Table 1. The survey was set up as an online questionnaire (Collaborators: https://goo.gl/forms/0ZzHtz9nokEeFx2r2, and Integrators: https://goo.gl/forms/b11SrpV7oGpnBdMX2).

Perceptions on SECO's Influencers

The population included 3,419 developers. We successfully sent the survey to 3,260 collaborators and had a response rate of 6.8% (224). The survey was also delivered to 159 integrators and had a response rate of 11.3% (18). The survey was sent on January 16th, 2018, and we received answers for 30 days. Participation was voluntary. Estimated time to complete the questionnaire was 10-15 minutes. We performed the analysis of the results in two groups to identify possible differences on how they realize influence.

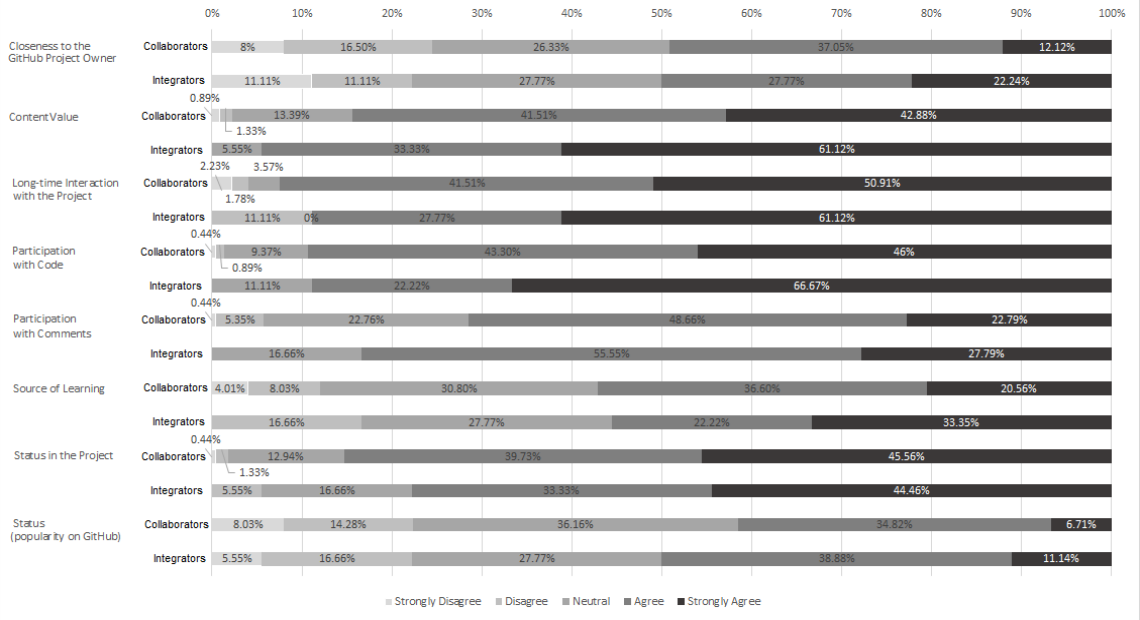
Collaborators' Opinions

In this group, 94.6% were male, 91.1% were professionals, 69.6% had Computer Science background, and 79% contributed to 5+ projects. According to collaborators, the most relevant characteristics to identify an influencer within a SECO are related to the project and to real collaboration (code/comments). "Long-time Interaction with the Project", "Participation with Code Contribution", and "Status in the Project" got the highest levels

of agreement (Figure 1). In a 1-10 range, 52.3% considered themselves as having levels of influence from 1 to 5, 10.7% selected 6, 11.6% selected 7, 15.6% selected 8, 4% selected 9, and 5.8% selected 10.

After analyzing the whole dataset, the top three definitions (that comprise others') to describe an influencer, according to collaborators, were: "*people who propose ideas, make PR's* (pull requests), *help reporting issues etc.*" (Collaborator #74), "*developers who take ownership of issues, they contribute with both ideas and code and help fix that code when it breaks*" (Collaborator #29), and "*other contributors, of code, but also of documentation, support, design and editorial work*" (Collaborator #95). Collaborators tend to focus on actions related to contributions instead of focusing on time and regularity they occur. They seem to see code contribution as important as comments, ideas or documentation, i.e. an influencer is responsible for some contribution actions.

FIGURE 1. Results for TQ-A – Collaborators' and Integrators' Opinions



Integrators' Opinions

In this group, 94.4% were male, 94.4% were professionals, 66.7% had Computer Science background, and 77.7% contributed to 5+ projects. Integrators' opinions were not divergent from collaborators' regarding identifying an influencer based on the project and collaborations performed by an influencer. "Participation with Code Contribution", "Content Value", and "Long-time Interaction with the Project" got the highest levels of agreement (Figure 1). Surprisingly, not all integrators considered themselves as having a high level of influence in npm SECO's projects: 50% declared having 1 to 5 level of influence, 11.1% selected 6, 22.2% selected 7, 11.1% selected 8, and 5.6% selected 10.

The top three definitions (that comprise others') to describe an influencer, according to integrators, were: "*active users (which write code/comments and offer/implement ideas)*" (Integrator #3), "*anyone who provides high quality constructive criticism across multiple issues*" (Integrator #12), and "*it depends on the projects. Usually some people who are using this project a lot and want to improve it by replying to github issues or opening PRs. Rarely some people even become regulars that help me triage issues and investigate*

bugs, when that's the case I give them collaborator access" (Integrator #15). Integrators consider contribution actions as relevant for someone seeking to become an influencer, but contribution should come with regularity in long-time interaction with the project.

Lessons Learned

Results show that contribution with comments and code is relevant to help identifying an influencer in an ecosystem. This seems to be so important that one integrator declared: "*when developers become regular at interacting with the project, it is possible that this collaborator receives a collaborator's status within this project*". Since "Status in the Project" was also listed as a relevant characteristic, a collaborator who participates in a project to become an integrator tends to make his/her influence even bigger in a SECO.

"Long-time Interaction with a Project" justifies why an integrator considers someone as an influencer. However, it seems that only contributing with a SECO's project for a long time is not enough: (1) contribution should be regular, and (2) it should not only be technical (code), but also social (comments, issues opening etc.).

We found out that being an influencer refers specifically to a SECO's project, and not to the performance/status a developer might have in other SECO's projects (or GitHub repositories), as "Status (popularity on GitHub)" got less agreement amongst developers. Influence may be connected to projects in a more individual sense. Moreover, "Closeness to the GitHub Project Owner" got higher level of rejection, which can indicate that it does not present much relation to influence, according to developers' perspective.

Finally, we believe that our analysis can be useful to integrators who are responsible for managing FLOSS projects in ecosystems. They are supported with a set of verified characteristics to identify developers who will probably influence the future of the SECO's project development as well as decision making and level of contribution.

Key Findings

Collaborators and integrators have very similar feelings on what characterizes an influencer in a SECO based on GitHub repositories. A *collaborator* who wants to be an influencer in a SECO's project starts by contributing with code/comments on other developers' contributions, and/or by opening issues and providing ideas. If the goal is to become not only an influencer but also an *integrator*, contribution should be regular since integrators would give the same privileges they have to other developers. Therefore, they could act as a co-developer towards becoming a core developer. An integrator should pay attention to collaborators that contribute regularly.

If the goal is to influence a project, the recipe is not much different from the collaborators': contribution actions and regularity. A SECO's project manager should be alert to influencers who are emerging from each project either to repeal bad influence (that could lead to unpleasant experiences) or to enhance good influence (to recruit others to make their ecosystem attractive to social developers). A useful strategy is to monitor discussions considering not only collaborators who eventually make code contributions, but also those who provide ideas/comments on other developers' issues and code.

Limitations and Challenges

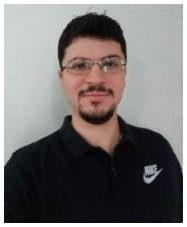
Some issues were not covered by our work and led to challenges. We did not consider the financial aspect (whether a developer is volunteer or paid[11]) and we focused on information that could be extracted from GitHub projects, which means that studying influence in other ecosystems could reveal new perspectives regarding this subject.

One challenge is to verify differences when influence is analyzed considering distinct aspects. For instance, developers can be paid to complete challenges and "copilot/helped" by others, and receive feedback from reviewers. It would be interesting to understand if the sense of influence changes when developers engage to win money or only to learn.

Another challenge is to verify how influence is recognized in platforms that have different social tools than GitHub. StackOverflow, Reddit and Hacker News could be investigated since they have interesting social functionalities, e.g. upvote or downvote a suggestion. Finally, a tool to support prediction of influencers as well as the analysis of different aspects of influence is of great importance for practitioners exploring the ecosystem perspective in the future of software development.

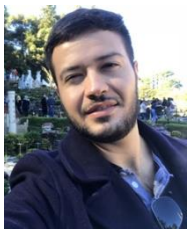
References

1. M. Lungu et al. The Small Project Observatory: Visualizing Software Ecosystems, Science of Computer Programming 75(4), 2010.
2. E. Constantinou, T. Mens. An empirical comparison of developer retention in the RubyGems and npm software ecosystems, Innovations in Systems and Software Engineering 13(2-3), 2017.
3. Cambridge Dictionary. https://dictionary.cambridge.org/pt/dicionario/ingles/influence.
4. G. Valença, C. Alves. Understanding How Power Influences Business and Requirements Decisions in Software Ecosystems, In ACM SAC, 2016, 1258-1263.
5. V. Farias et al. A Systematic Mapping Study on Influence in Software Ecosystems. Technical Report 12/2017-UNIRIO, 2017.
6. J. Carver et al. Distributed Teams, Developer Participation, and More, IEEE Software 34(3), 2017.
7. K. Crowston et al. Effective Work Practices for Software Engineering: Free/Libre Open Source Software Development, In ACM WISER, 2004, 18-26.
8. K. Manikas. Revisiting Software Ecosystems Research: A Longitudinal Literature Study, JSS 117, 2016.
9. B. Kitchenham et al. Evidence-Based Software Engineering and Systematic Reviews. Chapman and Hall/CRC, 2015.
10. R. Meloca et al. Understanding the Usage, Impact, and Adoption of Non-OSI Approved Licenses. In MSR, 2018, 11p.
11. G. Pinto et al. Who Gets a Patch Accepted First? Comparing the Contributions of Employees and Volunteers, In ACM/IEEE CHASE, 2018, 110-113.



Victor Farias is an Information Systems student at the Federal University of the State of Rio de Janeiro (UNIRIO), where he researches Software Ecosystems. Farias is a DevOps Engineer. Contact him at victor.farias@uniriotec.br.

Igor Wiese is an associate professor in the Department of Computing at the Federal University of Technology - Paraná, where he researches Mining Software Repositories techniques, Human Aspects of Software Engineering and related topics. Wiese received a PhD in computer science from the University of São Paulo. Contact him at igor@utfpr.edu.br.



Rodrigo Santos is an associate professor in the Department of Applied Informatics at Federal University of the State of Rio de Janeiro (UNIRIO), where he researches Software Ecosystems modeling/analysis, Requirements Engineering and related topics. Santos received a PhD in computer science from COPPE/UFRJ and participated in industrial P&D projects. Contact him at rps@uniriotec.br.

Highlights:

- Collaborators and integrators have very similar feelings on what characterizes an influencer in a software ecosystem based on GitHub repositories.
- A collaborator intending to become an influencer in a software ecosystem should start by contributing not only with code but also by opening issues and providing new ideas.
- A collaborator who is an influencer in a software ecosystem should have regular contributions to become a project integrator.