Introducing Agile Product Owners in a FLOSS Project

Matthias Müller, Christian Schindler, Wolfgang Slany

Introducing Agile Product Owners in a FLOSS Project

Matthias Müller, Christian Schindler, and Wolfgang Slany

Institute of Software Technology, Graz University of Technology, Austria
{mueller,cschindler,wslany}@ist.tugraz.at

Abstract. Sponsored Open Source Software projects, driven by various actors, have to balance the needs of volunteer contributors and business objectives. This work presents Catrobat, a FLOSS project established at Graz University of Technology, and how it introduced agile product owners. Product owners communicate the product vision, provide a general direction, decide about features, and prioritize requirements that are implemented by the community, i,e., they are ultimately responsible for the product. This agile approach is intended to ensure a certain outcome, such as business objectives, but also to react to the needs of community members and users on a short-term basis. This paper presents how therefore this role has been defined and the processes have been adapted.

Keywords: FLOSS · Open Source Software · Agile Software Development · Sponsored Open Source Communities.

1 Introduction

Governing Free/Libre Open Source Software (FLOSS) projects has already been subject of extensive research in the past. Publications on FLOSS [10] pointed out the role of leaders in such communities, that are either fulfilled by the projects' initiators or core-contributors. A main task for them is to provide a shared vision and govern the projects [7, 10]. However, in recent years we observed an increasing involvement of businesses in open source communities. Open source projects are nowadays often situated in ecosystems strongly connected to companies and social communities [5]. This means that businesses and other entities are establishing new, so called sponsored, open source projects [17]. These project not only follow a shared vision, but also business objectives set by the establishing entity. The challenge in governing these projects is to find a balance between the motivation of the contributors and business objectives [12]. Governance must therefore consider several aspects such as decision making, interaction, or release planning [5]. Failing in governing this relation between business objectives and community interests has been identified as a potential reason for the collapse of open source projects [1]. In general, leaders must be trusted by the community [7]. Therefore, governing open source projects becomes increasingly challenging and needs to respect a variety of different aspects.

In this work we present the case of Catrobat, a FLOSS project at Graz University of Technology, developing mobile visual programming frameworks and tools [15]. Although it is based on an open community, the main contributors are students who may participate as part of their curriculum [8]. These students are limited in the freedom of contribution, since a certain academic outcome is mandatory and there is also connected research with certain requirements that must be satisfied. In addition to that, the created software is published on different markets that require certain standards to be fulfilled. Requirements, originating independently of the contributors and users, e.g., by research or cooperation, must be balanced with the community's interests to keep them actively involved. Therefore, the project's leaders act as agile product owners since 2018. In this work we line out the process how this role was introduced in this project and how the resulting connected up- and downsides are dealt with.

2 Motivation to Introduce Product Owners

An all-time issue within the project has been that the number of proposed issues, representing user-stories and bugs, grew faster than the number of solved issues. Therefore, the need of prioritization came up to ensure that urgent and important requirements get finished in time. Furthermore, contributors did not thoroughly maintain the publicly reported issues, e.g., bugs or missing functionalities, resulting in a constantly growing issue pool. As also outlined in previous research [4], features requested by external parties often get unrecognized by core-developers. Lacking to meet requests from users can be frustrating for the community and may impact the project negatively [1]. These aspects made it necessary to introduce a role to keep track of, sort, prioritize issues independently from their origin. Catrobat already applies several individual agile methods and various chances and challenges of these methods were already discussed in the past [2, 3, 9]. Due to the positive experiences with agile principles, also this new role was supposed to be based on agile methodologies.

3 Product Owner within Catrobat

The introduction of product owners required several organizational changes. Besides defining this formal role, also processes and communication needed to be adapted to the new circumstances, as outlined in the following sections. Although this role is common in industry, special attention is needed in open communities such as Catrobat. Their character requires to focus on the balance between the different needs of contributors, users and external stakeholders, that are all involved for different individual reasons. Also the constant change of the community and direct involvement of users comes along with further challenges.

3.1 The Role "Product Owner"

Leaders in open source projects implicitly perform actions and fulfill responsibilities as they are described for product owners. In Scrum, a product owner

has the responsibility for the backlog to maximize value and represent external interests [6, 13]. Furthermore, they need to communicate the vision of the desired product and be a leader for the team [11]. It is important to note that these responsibilities are based on collaboration with the team, making it necessary to have a common understanding and language [13]. However, decisions by the product owner must be made visible to and be respected by all people involved [14]. *"The product owner is the one person ultimately responsible for the success or failure of the project"* [6]. Therefore, the founder of the project and experienced contributors have been assigned with this role. Although, Scrum defines this role for a single person [11, 14], these product owners form a board, similar to a committee that is common for FLOSS projects [7]. Derivations of Scrum, also having multiple product owners, can be found in successful industry projects too [16]. The decision therefore was based on the constant need of having a product owner available to the contributors, since previous research has shown contributors are working on an irregular schedule [9]. Therefore, constant availability for information exchange must be ensured. This has been identified as a main success factor for Scrum in industry [16]. Furthermore, whereas in industry this role should be performed as full-time position [11], in Catrobat, for a lack of resources, this role can just be fulfilled on a part-time basis, resulting in the need for several people.

Specific parts of the project also have *project owners* that are long-term and experienced contributors, having the same responsibilities and possibilities as the product owners within their specific scope. However, they are not allowed to develop user stories that they have specified themselves. This shall foster collaboration between all involved contributors. The co-structure of product owners representing sponsored goals, e.g., by research, cooperation or user-feedback, and projects owners originating from the community shall increase the commitment to the project. Introducing a governing structure that pays respect to both sides can be considered a main task for such open source communities [17].

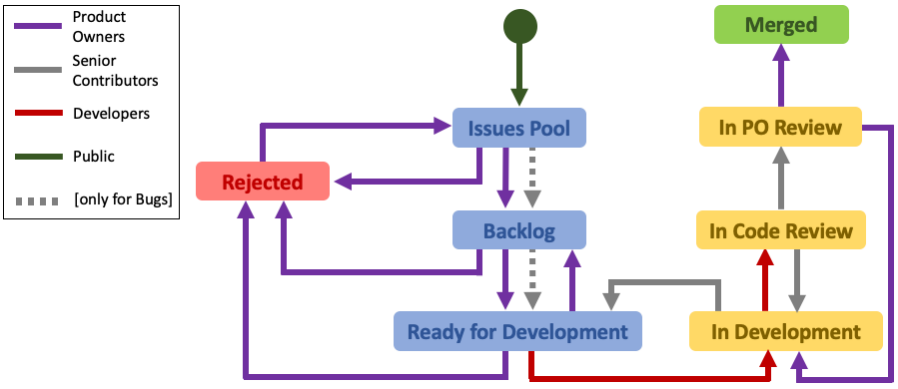

Fig. 1. Catrobat's development workflow considering product owner interactions.

3.2 Development Workflow

To introduce this new role, a development process, as illustrated in Figure 1, needed to be introduced. Catrobat's development is managed through a issue tracking system, which is open for all interested contributors. Also non-programmers are enabled by such issue tracking systems to report bugs and feature requests, which reflect the ideas of the users [4]. To prevent duplicate work and ensure the quality standards are met also external contributors are invited to work with this system. Besides that, this workflow is intended to allow frequent and fast releases, which is a challenge for many community driven projects [17]. Therefore, product owners in this workflow have three major tasks:

- Defining and prioritizing requirements and issues for the developers. Whereas, external contributors are not necessarily required to follow these predefined issues (e.g. work freely on ideas), participating students have to choose issues provided in *Ready for development*. However, also common contributors are asked to communicate their ideas to avoid rejection in the acceptance phase. Therefore, the creation of new issues is open for the public, also allowing to consider these ideas in the workflow, avoiding the mentioned rejection afterwards and foster involvement of users and the community.
- Discussing proposed requirements in planning games to clearly communicate the objectives and to get the developers' commitment .
- Functional acceptance of issues and merging them into the main repository. This step is necessary for all contributors to ensure the quality and prevent unfinished, buggy or inappropriate work being published.

The transition of issues from *Backlog* to *Ready for development* happens in a joint planning game. In this, the issues are estimated and developers assure a joint understanding of them. Whereas product owners have the key role in the first and last phase of the process, development is managed entirely by the developers. This includes that issues are not preassigned to contributors, reducing dependencies on certain individuals. Therefore, discussing proposed requirements with the developers and getting their commitment is essential for this agile workflow. Developers are also asked to review the work of others if it complies to the project's quality standards. This shall strengthen the collective code ownership in this process. An exemption in the workflow exists for bugs. They can be claimed by developers at any time without the involvement of product owners. Although, final product owner approval must be granted just like with scheduled issues. This supports the benefit of open source software projects - that bugs can be found, fixed and released quickly.

3.3 Communication

An important requirement for the introduction of product owners has been strengthening communication within the project. This step also focused on encouraging exchange between contributors, e.g., jointly discussing development tasks. Regular on-site meetings with student-contributors get reinforced and

Slack got introduced as main communication platform for all contributors, stimulating discussion in different topic specific channels. This is also intended to document decisions and information for currently absent contributors that might be involved in the following implementation phase. Beside the communication within the team, also product owners need regular meetings and exchange about the project's status. Therefore, following activities have been put into focus:

- Weekly Get-Together of to discuss upcoming features and requirements. This also includes backlog refinement and obtaining feedback from contributors.
- Monthly Planning-Games of product owners with the development team to schedule issues for *Ready for development*. During this event also the specification of issues is discussed, e.g., if developers need further clarifications, or if they are blocked by each other. Product owners hand over the prioritized issues for the next month to the contributors.
- Continuous exchange about current issues on designated Slack-channels and on an individual basis, e.g., via e-mail or comments on Jira and GitHub.

All this is intended to be open and transparent, what is essential for successfully governing FLOSS projects [7]. A challenge is the efficient communication between product owners. Especially for the case when one product owner answers questions from contributors, it is important that all others are informed about decisions made in their absence. This makes documentation and communication essential for this multi-person product owner approach.

4 Discussion

Although this workflow has just been introduced in early 2018, positive feedback is received from contributors. A challenge identified is to centralize communication that got essential for the collaborative nature of the workflow. Whereas contributors before exchanged in a way preferable for them, they must now be streamlined on dedicated channels. This is also true for the introduced product owners. An increasing number of messages and online time by contributors is outlining a preferable progress to tackle this challenge. Since this work solely points out the experience of this specific case, further long-term research on this approach is needed to be able to evaluate its success.

5 Conclusion

This work provides an approach for sponsored open source projects to balance the involved parties' needs and the freedom of contributors in an agile way. A simple workflow with clear responsibilities is provided that might be a response to the growing involvement of businesses or public institutions in open source communities. The introduced role of product owners can be seen as an extension to the already often defined role of leaders governing a FLOSS project. However, by following the proposed workflow and role definition based on Scrum, collaboration and communication in this open setting can be fostered, by ensuring the development of required business objectives at the same time.

References

1. Ehls, D.: Open source project collapse–sources and patterns of failure. In: Proceedings of the 50th Hawaii International Conference on System Sciences (2017)
2. Fellhofer, S., Harzl, A., Slany, W.: Scaling and internationalizing an agile foss project: Lessons learned. In: IFIP International Conference on Open Source Systems. pp. 13–22. Springer (2015)
3. Harzl, A.: Combining foss and kanban: An action research. In: IFIP International Conference on Open Source Systems. pp. 71–84. Springer (2016)
4. Heppler, L., Eckert, R., Stuermer, M.: Who cares about my feature request? In: IFIP International Conference on Open Source Systems. pp. 85–96. Springer (2016)
5. Kilamo, T., Hammouda, I., Mikkonen, T., Aaltonen, T.: From proprietary to open source—growing an open source ecosystem. Journal of Systems and Software **85**(7), 1467–1478 (2012)
6. Lacey, M.: The scrum field guide: Practical advice for your first year. Addison-Wesley Professional (2012)
7. Lerner, J., Tirole, J.: Some simple economics of open source. The journal of industrial economics **50**(2), 197–234 (2002)
8. Müller, M., Schindler, C., Slany, W.: Engaging students in open source: Establishing foss development at a university. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
9. Müller, M.: Agile challenges and chances for open source: Lessons learned from managing a floss project. In: 2018 IEEE Conference on Open Systems (ICOS). pp. 1–6 (2018)
10. Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., Ye, Y.: Evolution patterns of open-source software systems and communities. In: Proceedings of the international workshop on Principles of software evolution. pp. 76–85. ACM (2002)
11. Pichler, R.: Agile product management with scrum: Creating products that customers love. Addison-Wesley Professional (2010)
12. Rosén, T.: Open Source Business Model : Balancing Customers and Community. Ph.D. thesis, Linköping UniversityLinköping University, Industrial Marketing and Industrial Economics , The Institute of Technology (2008), report code: LiU-TEK-LIC 2008:26.
13. Schwaber, K.: Agile project management with Scrum. Microsoft press (2004)
14. Schwaber, K., Sutherland, J.: The scrum guide. Scrum Alliance (2017)
15. Slany, W., Luhana, K.K., Müller, M., Schindler, C., Spieler, B.: Rock bottom, the world, the sky: Catrobat, an extremely large-scale and long-term visual coding project relying purely on smartphones. In: Constructionsim 2018 (2018)
16. Sverrisdottir, H.S., Ingason, H.T., Jonasson, H.I.: The role of the product owner in scrum-comparison between theory and practices. Procedia-Social and Behavioral Sciences **119**, 257–267 (2014)
17. West, J., O'Mahony, S.: Contrasting community building in sponsored and community founded open source projects. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. pp. 196c–196c (2005)