How is your community doing?

Damian A. Tamburri
Patricia Lago
and Hans Van Vliet
Department of
Computer Science
VU University Amsterdam, The Netherlands
[d.a.tamburri, p.lago, j.c.van.vliet]@vu.nl

Martin von Weissenberg
Hanken School of Economics
Department of Information Systems
Helsinki, Finland
mvw@iki.fi

Abstract—**Engineering software is a social activity. Software projects' success is very often connected to how their development community fares. This connection becomes especially delicate when multiple communities have to join forces (e.g. in global software engineering). So far however, software engineering discipline lacks systematic methods to sense a community state to act upon. In this paper we propose a method to sense the state of a development community, by making its key attributes explicit. In addition, we suggest ways in which practitioners can use the community state, to adjust the relative communities (e.g. by adding specific technologies or changing organisational goals). One key finding is that certain barriers inhibit community effectiveness. These barriers can be tackled by analysing the state of the software development community. We illustrate our method with a case-study.**

I. Introduction

The craft of software is becoming more and more a social activity [**?**]. Several studies suggest that engineers spend between 70 and 85% of their time working with other people (e.g. end-user focus groups, managers, business sponsors, etc.) [1]. This means that in 70-85% of times, software engineering success is a community effort, rather than a success single individuals. This figure becomes more impactful with the increase of diversity and size of the development community. For example, Global Software Engineering is a development strategy entailing global teams to collaborate together from different locations in different timezones [2]. In this circumstance, time, space and culture distances magnify the risks of project failure connected to failures in the development community [3], [4], [5]. The problem persists since the software engineering practice still lacks a systematic method to identify the state of development communities. As a consequence, there is no way for software engineering practitioners to effectively act upon the development community they are part of. To tackle this issue, this paper proposes a method to "sense" and manipulate a development community's status. The method is applied to a real-life organisation. We found that the method helps uncovering the state of development communities but it cannot be automated.

We found that observable *social communities* are a composition of known social community types [5], [6]. Each type can be characterised with attributes that are necessary and sufficient to uniquely identify it [**?**]. We argue that the status of a community is given by the (set of) values of attributes that are necessary and sufficient to identify (all) its components.

First, our method uses a questionnaire to identify the current state of the observed community. Second, our method uses a decision-tree to uncover the communities represented by the organisational information obtained. Finally, we compare the identified social community attributes (i.e. the community state) with empirical results from previous work [6]. This last step allows us to understand what can be done to improve the communities, e.g. by encouraging changes in its attributes. The benefits of using our method are threefold: first, practitioners can use the questionnaire as a checklist for missing management information; second, the questionnaire can be used as a guideline to engineer communities for a specific software development problem; third, the community state we obtain can be cross-referenced with development velocity to evaluate community effectiveness in the current state. The rest of the paper is structured as follows: Section II provides some background on software engineering and the study of its social aspects; Sections III and IV provides our research questions, evidence and methods; Section V provides or results; Section VI presents the case-study we conducted to validate our work. Finally, Sections VII and VIII discuss our results, case-study and conclude the paper.

II. On the Benefits of Studying and Supporting Software Development Communities

III. Research Questions

The work presented in this paper was realised rotating around three research questions.

First, **how can we identify the status of a software development community?** .

Second, **how can we act on the status of a software development community?** .

Finally, **how can practitioners use the state-of-the-art in social communities to improve development collaboration?** .

IV. Evidence Used and Methods

A. Previous and Related Work

B. Materials Used

C. Research Methods

V. Results

VI. Sensing Community state: a Case-Study

VII. Discussion

VIII. Conclusion

Acknowledgment

The authors would like to thank beer, Mirella's home-made sweets and her mother's home-made cheese.

References

[1] J. Keyes, *Social software engineering :*. Boca Raton, FL :: Taylor & Francis, Auerbach Series, 2011.

[2] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish, and J. Kazmeier, *Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series)*. Boston, MA, USA: Auerbach Publications, 2006.

[3] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "An empirical study of global software development: distance and speed," in *ICSE '01: Proceedings of the 23rd International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 81–90.

[4] N. Nagappan, B. Murphy, and V. Basili, "The influence of organizational structure on software quality: an empirical case study," in *International conference on Software engineering*. Leipzig, Germany: IEEE, May 2008, pp. 521–530. [Online]. Available: http://doi.acm.org/10.1145/1368088.1368160

[5] D. A. Tamburri, E. di Nitto, P. Lago, and H. van Vliet, "On the nature of the gse organizational social structure: an empirical study," *International Conference on Global Software Engineering*, 2012.

[6] D. A. Tamburri, P. Lago, and H. van Vliet, "Organizational social structures for software engineering." Available upon request, under submission, 2012, pp. 1–35.