Human Aspects, Gamification, and Social Media in Collaborative Software Engineering*

Bogdan Vasilescu
Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
b.n.vasilescu@tue.nl http://www.win.tue.nl/~bvasiles

ABSTRACT

Software engineering is inherently a collaborative venture. In open-source software (OSS) development, such collaborations almost always span geographies and cultures. Because of the decentralised and self-directed nature of OSS as well as the social diversity inherent to OSS communities, the success of an OSS project depends to a large extent on the social aspects of distributed collaboration and achieving coordination over distance. The goal of this dissertation research is to raise our understanding of how human aspects (*e.g.*, gender or cultural diversity), gamification and social media (*e.g.*, participation in social environments such as Stack Overflow or GitHub) impact distributed collaboration in OSS.

Keywords

Collaborative software engineering, open source.

Categories and Subject Descriptors

H.5.3 [**Information interfaces and presentation (e.g., HCI)**]: Computer-supported cooperative work.

General Terms

Human Factors

1. INTRODUCTION

Software engineering is inherently a collaborative venture, involving many software engineers that coordinate their efforts to produce a large software system [23, 38]. In open-source software (OSS) development, such collaborations almost always span geographies and cultures, although globally distributed projects are becoming the norm for large commercial software systems as well [15]. Yet, although decentralised and self-directed, OSS development yields high quality contributions rivalling commercial work produced by geographically collocated teams under close supervision [26].

In OSS, developers, supporters, and users, likely with different skill sets and skill levels, different personalities, different geographic locations and different cultural backgrounds *organise themselves* in online communities and *voluntarily contribute* to a collaborative software project [23]. Therefore, because of the decentralised and self-directed nature of OSS as well as the social diversity inherent to OSS communities, the success of a project hinges to a greater extent than in commercial settings on the social aspects of distributed collaboration and achieving coordination over distance. To facilitate distributed collaboration, online environments targeting software developers, such as GitHub (hosting code repositories) or the Stack Exchange network of question and answer websites (*e.g.*, Stack Overflow), have emerged and gained popularity in recent years.

Borrowing from ecology, in this thesis I view OSS communities as knowledge-sharing *ecosystems* [22,27]: software developers (the organisms; biotic components) interact among themselves and with the environment (software and hardware tools; abiotic components); these biotic and abiotic components are linked together through *knowledge flows* in a symbiotic relationship, wherein "the community learns from its participants, and each individual learns from the community" [27]. In a similar light, emerging collaborative environments such as GitHub or Stack Overflow can be seen as *disturbances* to knowledge-sharing ecosystems surrounding traditional OSS development: due to their many social media features or their use of gamification [10], they may have profound and immediate effects on an ecosystem and the knowledge flows therein, changing the ways in which developers collaborate, learn, and communicate among themselves and with their users [3,9,28]. These disturbances may prove either beneficial (*e.g.*, Stack Overflow is known to provide good technical solutions [25] and to provide them fast [21]) or harmful to the ecosystem (*e.g.*, social media may lead to interruptions impairing the developers' performance [28]). Their effects are thus far not well understood. My dissertation research seeks to raise our understanding of these effects by answering:

> How are OSS knowledge-sharing ecosystems responding to *disturbances* such as GitHub or Stack Overflow and what influences their response?

2. APPROACH AND PROGRESS

To address this research question, together with my team I explored trace data publicly available in OSS. For example, most OSS projects record historical activities of their contributors in version control systems, issue trackers, or

*The paper describes a mature stage of work.

ICSE '14, May 31 – June 7, 2014, Hyderabad, India

mailing lists, while data dumps from both GitHub [13, 14] and Stack Overflow are also available for mining.

So far, this wealth of OSS trace data has enabled my team and me to explore two knowledge-flow paths (interactions between *knowledge seekers* and *knowledge providers* via a *medium* through which knowledge flows). Our endeavours have resulted in a series of publications (detailed below) describing empirical studies of various OSS communities, including Gnome, a popular desktop environment for UNIX-type operating systems, Emacs, a popular text editor, R, a popular data analysis software as well as GitHub and Stack Overflow. The first series of studies (Section 2.1) explored diversity in traditional OSS ecosystems such as Gnome, and can be considered as a baseline for our subsequent studies. The second series of studies (Section 2.2) explored activity in GitHub and Stack Overflow and contrasted that to more traditional communities.

2.1 Knowledge Sharing through Contributions to Source Code Repositories.

Background. One of the knowledge-flow paths involves developers (knowledge providers) sharing their knowledge through contributions to the source code repositories (the medium), with the goal of driving further the evolution of the software project (knowledge consumer) and ensuring the ecosystem's sustainability.

Given the natural diversity of human beings, one can also expect heterogeneity of knowledge providers within an OSS ecosystem. Some contributors are likely to be considerably more active than others; some will possess different skills than others; not all will be involved in the same activities (*e.g.*, coding, translating, writing documentation). Stated differently, OSS communities are socially and technically diverse, with contributors communicating, interacting and collaborating differently. The social interactions between OSS contributors, as well as their degree of project participation have been reported repeatedly to influence software quality and complexity (*e.g.*, [4]). Social diversity in technical teams is also known to influence the team's performance [24].

Progress. In two empirical studies of contributors to Gnome [34] and Emacs [37] code repositories, we uncovered:

- how workload and involvement of ecosystem contributors vary across projects and across activity types [34];
- the extent to which contributors specialise in particular activity types (e.g., writing code vs. contributing to localization) [34];
- how the differences in skills between contributors can be modelled, and what impact such differences might have on the ecosystem's sustainability [37].

Outlook. At the very least, the two empirical studies allowed us to confirm that there is no such thing as a uniform ecosystem of projects and contributors. For instance, when taking into account the activities a contributor is involved in (*e.g.*, coding, localization, or contributing to documentation), there is a lot of variation across developers in terms of their workload (*i.e.*, amount of contributions): more technically diverse contributors (*i.e.*, those involved in many different activity types) tend to be more productive [34].

Future work will also explore the social diversity direction (*e.g.*, gender, age, cultural background) and relationships between both social and technical diversity, on the one hand, and quality and quantity of contributions, on the other hand.

2.2 Knowledge Sharing through Questions and Answers on Social Q&A Platforms.

Background. Perhaps the most distinctive feature of social Q&A platforms (the most visible of which is Stack Overflow [21] for programming questions and answers) is *gamification*, *i.e.*, participants compete to obtain reputation points and badges, which enable additional privileges once various thresholds are exceeded (*e.g.*, moderation rights). Reputation points and badges on such sites can be seen as a measure of one's expertise by peers and potential recruiters [7], and are known to motivate users to contribute more [1, 10]. In this setting, developers can act both as knowledge seekers and as knowledge providers.

As knowledge seekers: Developers create software by standing on the shoulders of others [28]: they reuse components and libraries, and go foraging on the Web for information that will help them in their tasks [6]. Here, knowledge-flow paths are formed between developers (knowledge seekers) asking, *e.g.*, on Stack Overflow (medium) for help with their code, and the experts (knowledge providers) answering their questions.

As knowledge providers: Developers also engage in online support fora, such as mailing lists or Q&A platforms (media), as experts (knowledge providers), to satisfy a demand for knowledge. Here, knowledge seekers are other developers, perhaps less experienced, or users themselves.

Progress. To study the effects associated with participating in Stack Overflow for both these roles of developers (knowledge seekers and knowledge providers), I created two longitudinal data sets combining activity on Stack Overflow with activity in OSS, wherein participants overlap: one data set contains changes made to GitHub repositories [32]; the other contains communication on R mailing lists [33]. Using these data sets we found that:

- there is a positive connection between participation in Stack Overflow, on the one hand, and productivity (in terms of amount of contributions) on GitHub [32] and R mailing lists, on the other hand [33];
- activity (expertise) on one platform is indicative of activity (expertise) on the other [32];
- more active developers are more likely to engage in Stack Overflow [32, 33];
- some groups of developers benefit more from participation in Stack Overflow than others [32];
- the same knowledge providers answer questions faster in the gamified Stack Overflow environment than on mailing lists [33];
- Stack Overflow attracts mailing list experts, who disengage from mailing lists and transition to the new social medium [33].

Outlook. The two empirical studies combining activity on Stack Overflow with activity on GitHub [32] and on R mailing lists [33] indicate that developers are indeed attracted by gamified social environments such as the one offered by Stack Overflow. The access to expert knowledge has a positive influence on their productivity in OSS. At the same time, the increased visibility and recognition made possible by such environments as Stack Overflow motivate them to contribute more, *e.g.*, to the extent that they engage more on Stack Overflow than they do on mailing lists, or migrate entirely to the new environment. These ob-

servations were acknowledged by the developers themselves during a survey [33].

These findings seem to suggest that gamification and social media elements should be embraced in software engineering practices and tools. However, such mechanisms are not without criticism. For instance, in its current implementations gamification is almost synonymous with achievement, while achievement is not necessarily well suited as a motivator for many cultures around the world [16].

Similarly, gender differences also become apparent in competitive environments, where women tend to be less effective than men [12]. In an empirical study comparing activity on Stack Overflow with that on traditional mailing lists [30, 31], we tried to understand how and when women developers engage in software-development-related online communities. My team and I found, *e.g.*, that gamified platforms such as Stack Overflow are less successful at retaining women than traditional mailing lists, although for the shorter duration of their involvement women achieve activity levels comparable to those of men.

Based on these insights, future research will explore how gender and other human aspects such as cultural background could play a role when incorporating gamification and social media elements into the design of software engineering tools and the improvement of software engineering practices.

3. CONTRIBUTIONS

The empirical and methodological contributions resulting from this work (best labelled as collaborative software engineering) serve two research communities, computer-supported cooperative work (CSCW) and software engineering (SE), with important implications for both.

First, these results directly serve the OSS communities from which they were derived, by offering empirical evidence about which factors (*e.g.*, gender) influence the health and popularity of these communities and the engagement of users therein. Second, although not immediately generalisable beyond the OSS communities in which they have been derived, we expect these results to help inform the design of software engineering tools (*e.g.*, incorporating gamification features) and the advancement of software engineering practices. Third, this work has led to numerous methodological improvements (outlined below) benefiting the empirical software engineering community, or any other researchers analysing trace data. These techniques are also applicable to other domains, as we have shown, *e.g.*, in the metastudy of the "health" of software engineering conferences [35, 36].

3.1 Methodological Advances

Robust identity merging techniques. One of the biggest challenges when mining software repositories is identity merging [5, 18, 34]. Both within a single repository as well as across repositories, the same person may use different aliases, *i.e.*, different (*name, email*) tuples. The extent of the problem is unpredictable. In our study of Gnome [34] we found that one of the developers used 168 different aliases when committing changes to the source code repository throughout his involvement in the ecosystem. My group and I have developed and are refining a robust identity merging algorithm (described in [18]) based on Latent Semantic Analysis (LSA), a popular information retrieval technique, which performs very well in presence of noisy data, as well as a number of heuristics (described, *e.g.*, in [33,34]).

Integrating data from multiple repositories. Assembling joint data sets from multiple sources, in which participants are identifiable (*e.g.*, developers active simultaneously on GitHub and Stack Overflow, or on Stack Overflow and mailing lists) permits their activities to be linked across the different resources and also over time, enabling richer empirical studies. My team and I have been pioneers in linking activity on Stack Overflow with activity in OSS (*e.g.*, in [32,33]) using identity merging techniques adapted specifically for this purpose. This integration has also led to other methodological advances, such as correctly and carefully reconstructing discussion threads from email archives (described in [33]).

Portfolio of statistical techniques. When performing empirical studies in software engineering, one often tries to assess whether the distributions of a given metric are different for different population groups. Traditionally, comparison of multiple groups follows a two-step approach: first, a global null hypothesis is tested and then multiple comparisons are used to test sub-hypotheses pertaining to each pair of groups. Two-step approaches are often criticised (*e.g.*, the global test null hypothesis may be rejected while none of the sub-hypotheses are rejected, or vice versa [11]). My team and I have been the first to apply more modern and more robust one-step approaches [17] to software engineering data (*e.g.*, in [32, 34]), and to propose a visualisation for the results of this statistical procedure.

4. RELATED WORK

This section summarizes some milestone works which are not already referenced above. Popularity of StackOverflow among software developers has lead to increased interest from the research community as well [29]. Bacchelli *et al.* [2] and Cordeiro *et al.* [8] argue that the current lack of integration between Q&A websites and modern IDEs forces developers to interrupt their flow and change context every time they need to deal with them, thus delaying their activity. Xuan et al. [39] argue that social communication activities (such as asking or answering Stack Overflow questions) may delay programming activities, since both of these activities compete for the time resources of developers. Instead, Brandt et al. [6] propound that by relying on information and source code fragments from the Web, developers more effectively distribute their cognition, allowing them to devote more energy to higher-level tasks.

Both participation in Stack Overflow and software development in public GitHub repositories can be seen as social activities. Dabbish *et al.* [9] report that GitHub users are aware of being watched by their peers, and this awareness influences how they behave and construct their actions, for example, by making changes less frequently. Lee *et al.* [20] and Lakhani and von Hippel [19] discuss what motivates developers to contribute to social Q&A sites such as Stack Overflow, listing instances of what the economic literature calls "signalling incentives" (career concerns, desire for peer recognition, reputation building) as main reasons.

5. EVALUATION

Evaluation of the preliminary results outlined in this paper has been carried out following a mixed-methods approach. To achieve triangulation of our findings, quantitative (statistical) analyses have been complemented by case

studies, surveys and interviews. Future results emerging during this dissertation work will be evaluated in a similar fashion. In addition, it is well known that empirical studies are prone to numerous threats to validity. We are doing our best to reduce these threats by continuously refining our data collection methods and applying appropriate statistical techniques.

6. ACKNOWLEDGEMENTS

Many thanks to my PhD co-advisors, Alexander Serebrenik and Mark van den Brand, all my co-authors and the Dutch Science Foundation (NWO) for financial support through the research project NWO 600.065.120.10N235.

7. REFERENCES

[1] Anderson, A., Huttenlocher, D. P., Kleinberg, J. M., and Leskovec, J. Discovering value from community activity on focused question answering sites: a case study of Stack Overflow. In *KDD*, ACM (2012), 850–858.

[2] Bacchelli, A., Ponzanelli, L., and Lanza, M. Harnessing Stack Overflow for the IDE. In *RSSE*, IEEE (2012), 26–30.

[3] Begel, A., Bosch, J., and Storey, M.-A. Social networking meets software development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. *IEEE Software 30*, 1 (2013), 52–66.

[4] Bettenburg, N., and Hassan, A. E. Studying the impact of social structures on software quality. In *ICPC*, IEEE (2010), 124–133.

[5] Bird, C., Gourley, A., Devanbu, P. T., Gertz, M., and Swaminathan, A. Mining email social networks. In *MSR*, ACM (2006), 137–143.

[6] Brandt, J., Guo, P. J., Lewenstein, J., Dontcheva, M., and Klemmer, S. R. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *CHI*, ACM (2009), 1589–1598.

[7] Capiluppi, A., Serebrenik, A., and Singer, L. Assessing technical candidates on the social web. *IEEE Software 30*, 1 (2013), 45–51.

[8] Cordeiro, J., Antunes, B., and Gomes, P. Context-based search to overcome learning barriers in software development. In *RAISE* (2012), 47–51.

[9] Dabbish, L. A., Stuart, H. C., Tsay, J., and Herbsleb, J. D. Social coding in GitHub: transparency and collaboration in an open software repository. In *CSCW*, ACM (2012), 1277–1286.

[10] Deterding, S. Gamification: designing for motivation. *Interactions 19*, 4 (2012), 14–17.

[11] Gabriel, K. R. Simultaneous test procedures—some theory of multiple comparisons. *ANN MATH STAT 40*, 1 (1969), 224–250.

[12] Gneezy, U., Niederle, M., and Rustichini, A. Performance in competitive environments: Gender differences. *Q. J. Econ. 118*, 3 (2003), 1049–1074.

[13] Gousios, G. The GHTorrent dataset and tool suite. In *MSR*, IEEE (2013), 233–236.

[14] Gousios, G., and Spinellis, D. GHTorrent: Github's data from a firehose. In *MSR*, IEEE (2012), 12–21.

[15] Herbsleb, J. D. Global software engineering: The future of socio-technical coordination. In *FOSE*, IEEE (2007), 188–198.

[16] Khaled, R. It's not just whether you win or lose: Thoughts on gamification and culture. In *Gamification Workshop at CHI*, ACM (2011).

[17] Konietschke, F., Hothorn, L. A., and Brunner, E. Rank-based multiple test procedures and simultaneous confidence intervals. *Electron. J. Stat. 6* (2012), 738–759.

[18] Kouters, E., Vasilescu, B., Serebrenik, A., and van den Brand, M. G. J. Who's who in Gnome: Using LSA to merge software repository identities. In *ICSM*, IEEE (2012), 592–595.

[19] Lakhani, K. R., and von Hippel, E. How open source software works: "free" user-to-user assistance. *Research Policy 32*, 6 (2003), 923–943.

[20] Lee, S., Moisa, N., and Weiss, M. Open source as a signalling device - an economic analysis. Working Paper Series: Finance and Accounting 102, Dept Finance, Goethe Univ Frankfurt am Main, 2003.

[21] Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. Design lessons from the fastest Q&A site in the west. In *CHI*, ACM (2011), 2857–2866.

[22] Mens, T., Claes, M., Grosjean, P., and Serebrenik, A. *Evolving Software Systems*. Springer, 2014, ch. Studying Evolving Software Ecosystems based on Ecological Models.

[23] Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., and Ye, Y. Evolution patterns of open-source software systems and communities. In *IWPSE*, ACM (2002), 76–85.

[24] Neale, M. A., Northcraft, G. B., and Jehn, K. A. Exploring Pandora's box; the impact of diversity and conflict on work group performance. *Performance Improvement Quarterly 12*, 1 (1999), 113–126.

[25] Parnin, C., Treude, C., Grammel, L., and Storey, M.-A. Crowd documentation: Exploring the coverage and the dynamics of API discussions on Stack Overflow. Tech. rep., Georgia Tech, 2012.

[26] Raymond, E. S. *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*. O'Reilly, 2008.

[27] Sowe, S. K., Stamelos, I., and Angelis, L. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *JSS 81*, 3 (2008), 431–446.

[28] Storey, M.-A. D., Treude, C., van Deursen, A., and Cheng, L.-T. The impact of social media on software engineering practices and tools. In *FoSER*, ACM (2010), 359–364.

[29] Vasilescu, B. Academic papers using Stack Overflow data. `http://meta.stackoverflow.com/q/134495`.

[30] Vasilescu, B., Capiluppi, A., and Serebrenik, A. Gender, representation and online participation: A quantitative study of StackOverflow. In *Social Informatics*, ASE/IEEE (2012), 332–338.

[31] Vasilescu, B., Capiluppi, A., and Serebrenik, A. Gender, representation and online participation: A quantitative study. *Interacting with Computers* (2013).

[32] Vasilescu, B., Filkov, V., and Serebrenik, A. StackOverflow and GitHub: Associations between software development and crowdsourced knowledge. In *SocialCom*, ASE/IEEE (2013), 188–195.

[33] Vasilescu, B., Serebrenik, A., Devanbu, P. T., and Filkov, V. How social Q&A sites are changing knowledge sharing in open source software communities. In *CSCW*, ACM (2014), 342–354.

[34] Vasilescu, B., Serebrenik, A., Goeminne, M., and Mens, T. On the variation and specialisation of workload – A case study of the Gnome ecosystem community. *Empirical Software Engineering* (2013).

[35] Vasilescu, B., Serebrenik, A., and Mens, T. A historical dataset of software engineering conferences. In *MSR*, ACM (2013), 373–376.

[36] Vasilescu, B., Serebrenik, A., Mens, T., van den Brand, M., and Pek, E. How healthy are software engineering conferences? *Science of Computer Programming* (2014).

[37] Vasilescu, B., Serebrenik, A., and van den Brand, M. G. The Babel of software development: Linguistic diversity in Open Source. In *SocInfo*, Springer (2013), 391–404.

[38] Whitehead, J. Collaboration in software engineering: A roadmap. In *FOSE*, IEEE (2007), 214–225.

[39] Xuan, Q., Gharehyazie, M., Devanbu, P. T., and Filkov, V. Measuring the effect of social communications on individual working rhythms: A case study of open source software. In *Social Informatics*, ASE/IEEE (2012), 78–85.